COMMODORE DISC/64/128

COMMODORE DISC C64/C128

Nr. 35 DM 19,80 ÖS 168,– SFR19,80

Unverbindliche Preisempfehlung

24 Commodore Programme auf Disc im Heft

---

165 Kilo-Byte ohne abtippen

Alle Programme auf Disc im Heft!

C 64:

128 PC:

CD Intern

Hallo, liebe Leser!

Gleich zweimal haben Sie die Möglichkeit, etwas über Programmierung zu lernen. Peter Bergen erklärt in seinem dritten Teil der Folge „So programmiere ich in BASIC" die Programmierung von Action-Spielen. Wie Sie vielleicht bemerkt haben, hatten wir in der vorigen COMMODORE DISC etwas durcheinander gebracht und bereits die vierte Folge abgedruckt. Doch wollen wir Ihnen auch die dritte Folge nicht vorenthalten.
Action-Spiele sind ein schwieriges Thema. BASIC ist für hohe Geschwindigkeiten und besondere Effekte oft unzureichend. Dennoch gibt es Möglichkeiten. Die eine ist die Verwendung von Maschinenroutinen, die speziell für Action-Games existieren, und die von BASIC aus angesprochen werden können. Die andere ist, fehlende Effekte durch gute Ideen auszugleichen. Peter Bergen demonstriert letztere Möglichkeit und verzichtet gar auf Sprites. Ausführlich geht er dagegen auf Video- und Farbspeicher sowie auf die Erstellung von User-Zeichen ein. Für Action-Spiele ist dieses Grundlagenwissen sicherlich nicht zu entbehren.
Nochmals um Programmierung geht es bei Magic-Clock. Magic-Clock zeigt, daß Programme auf verschiedenen Rechnern laufen können, obwohl sie vielleicht nicht nur lupenreines BASIC V2.0 verwenden, sondern auch auf einige sich unterscheidende RAM- oder ROM-Adressen zugreifen. Doch das eigentliche Anliegen ist die Demonstrierung eines guten und strukturierten Programmierstils. Programme können so geschrieben werden, daß sie übersichtlich gestaltet und damit änderungsfreundlich sind. Man findet sich auch noch nach einiger Zeit in ihnen zurecht.
Programme können gut durchdacht sein. Wenn vor dem Codieren das Programmkonzept gut durchdacht wurde, läßt sich vieles vereinfachen. Wenn mehrmals vorkommende Abarbeitungvorgänge als Unterroutinen verfaßt werden, statt einer Vielzahl von Fallunterscheidungen kurze Berechnungen erfolgen, lassen sich erstaunliche Kürzungen erzielen. Das Programm Rubik's Clock aus COMMODORE DISC 29 benötigte 76 Blocks auf der Diskette. Eine C128-Version von Magic-Clock brachte es unter Weglassung der REM-Zeilen nur auf neun Blocks. Das ist weniger als der achte Teil. Magic-Clock zeigt vor allen Dingen, daß bei der Programierung zuerst mit treffenden Worten gearbeitet werden sollte, da wir uns darunter eine Menge vorstellen können. Erst zuletzt wird der eigentliche BASIC-Code notwendig. Das äußert sich im Programm darin, daß erst die Routinen auf unterster Ebene reinen BASIC-Code enthalten, zuvor sind es meist Aufrufe geschickt gewählter Routinen und Funktionen.
Erst nachdem ich den Ablauf vollständig mit eigenen Definitionsworten festgelegt hatte und damit zufrieden war, begann die Umsetzung in Programmiersprachen von FORTH über Pascal zu BASIC. Auf Flußdiagramme konnte vollständig verzichtet werden. Notwendig sind solche Hilfsmittel nur bei schlechter und unübersichtlicher Programmierung mit Monsterroutinen und verwirrenden Quersprüngen. Magic-Clock ist nicht nur ein zu lösendes Knobelspiel, sondern ein Programmier-Lehrstück, aus dem Sie sicherlich Nutzen für Ihren eigenen Programmierstil ziehen können.
Ein Leserbrief war Anlaß für unsere Sortierroutinen. Wer die Prinzipien der einfachen Sortierprogramme versteht, wird sich in Zukunft stets zu helfen wissen. Sollten die komplexeren Verfahren nicht ganz klar werden, so ist dies kein Beinbruch. Sie brauchen lediglich unsere Sortierroutinen in Ihre eigenen Programme einbinden. Welches Sortierprogramm für Ihre Zwecke das richtige ist, können Sie durch Vergleich der Sortierzeit und Anmerkungen im Begleitartikel entscheiden. In zwei Versionen, nämlich für Integerzahlen und Textstrings, liegen die Algorithmen vor. Falls Sie Real-Zahlen benötigen, dürfte das Umschreiben kein Problem bereiten.
Besondere Leckerbissen sind das Zeichenprogramm Hyprapaint und das Mathematikprogramm Mega-Maths I. Hypra-Paint ist dank Maschinensprache äußerst schnell, besitzt eine Menge von Optionen und eine ausgereifte Windowtechnik. Ein besonderes Merkmal ist, daß die Zeichenfläche nicht nur aus einem einzigen Bildschirm besteht. Der Bildschirm ist ein frei wählbarer Ausschnitt der Zeichenfläche, die eine ganze DIN-A4-Seite umfaßt. Dadurch können Sie ganz schön viel Grafik auf der Zeichenfläche unterbringen. Daß einzelne Punkte editiert werden können, daß die Grafik beschriftet werden kann, darüber brauchen wir nicht viele Worte verlieren, denn dies und anderes sind für Hypra-Paint Selbstverständlichkeiten.
Mega-Maths I ist ein sehr umfangreiches Mathematikprogramm. Es besteht aus einer Sammlung von Formeln und bietet eine Unzahl von Berechnungsmöglichkeiten. Außerdem gehört dazu die Möglichkeit, Funktionen grafisch am Bildschirm darzustellen. Da Mega-Maths I diese drei Teile zusammen enthält, ist es das Mathematikprogramm für fast alle Fälle. Jedem, der mit Mathematik konfrontiert wird, ist Mega-Maths I von Nutzen.
Der Fußball besitzt viele Freunde. Das beweisen die Einschaltquoten von Sportsendungen und die Mitgliederzahl von Fußballvereinen. Während bei Bundesligen die Fußballtabellen im Fernsehen erscheinen, müssen sie in anderen Ligen erst erstellt werden. Leage Manager Pro ist eine Liga-Verwaltung für jede Fußballiga von elf bis 22 Vereinen. Spielpaarungen und Spielergebnisse können eingegeben, gespeichert und abgefragt werden. Den Tabellenstand errechnet der Computer.
File-Manager 128 dient zum Verwalten Ihrer Programmsammlung. Programmverzeichnisse können manuell oder automatisch erstellt werden. Die alphabetische Sortierung übernimmt der Computer. Auf den Listen fehlt auch nicht der Hinweis auf die richtige Diskettenbox.
Rechnung und Mahnungen hat jeder Handel Treibende zu schreiben. Auch hierbei hilft der Computer. Für diejenigen, die ihr Glück gerne aus den Sternen erfragen, gibt es jetzt das Prognoseprogramm Horoskop, das Ihnen für jeden Tag eine Voraussage liefert. Ein weiteres Programm, Landstraße, ist superkurz. Vielleicht sollten Sie einmal versuchen, auch so etwas Einfaches zu programmieren.
Wenn Ihnen besondere Themen am Herzen liegen, über die Sie gerne etwas mehr erfahren würden, so schreiben Sie uns. Für Anregungen sind wir immer dankbar.

Ihr
Alfons Mittelmeyer

INHALT

AUF DISC IM HEFT


```
0 "commodore disc35" 35 2a
1    "boot.64"          prg
10   "disclader64/128"  prg
0    "================" del<
0    "     c-64       " del<
0    "================" del<
124  "horoskop"         prg
16   "magic clock"      prg
0    "----------------" del<
3    "sort-info"        prg
2    "hash-sort"        prg
2    "select-sort.int"  prg
2    "select-sort.txt"  prg
2    "bubble-sort.int"  prg
2    "bubble-sort.txt"  prg
2    "insert-sort.int"  prg
2    "insert-sort.txt"  prg
3    "quick-sort.int"   prg
3    "quick-sort.txt"   prg
4    "turbosort.int"    prg
4    "turbo-sort.txt"   prg
0    "================" del<
0    "     c-128      " del<
0    "================" del<
2    "bubble-128.int"   prg
2    "insert-128.int"   prg
3    "quick-128.int"    prg
4    "turbo-128.int"    prg
0    "----------------" del<
22   "paddles"          prg
1    "bin-dez"          prg
0    "----------------" del<
194  "mega-maths i"     prg
12   "graph-i"          prg
0    "----------------" del<
55   "league manager"   prg
3    "lm.datas 1"       prg
8    "lm.datas 2"       prg
0    "----------------" del<
34   "rechn & mahn"     prg
60   "hyprapaint"       prg
75   "file-manager 128" prg
4    "landstrasse"      prg
0    "================" del<
0    "(c) copyright by" del<
0    "ca - verlag gmbh" del<
0    "================" del<
2 blocks free.
```


Wichtiger Hinweis! *Die Diskette in diesem Heft ist weder list- noch kopiergeschützt. Aus verständlichen Gründen können wir daher bei Programmfehlern lediglich Umtauschrecht einräumen. Das Rückgaberecht gegen Kaufpreiserstattung ist ausgeschlossen! Sollte also eines der Programme nicht laufen, senden Sie die Original- Diskette an den Verlag zurück. Sie erhalten selbstverständlich eine korrigierte Fassung. Anschrift: Siehe Impressum.*

WORKSHOP

SERVICE

Magic-Clock

Ein Lehrstück

Das Programm Rubik's-Clock für C64, C128, C16 und P4 ist nicht nur als Spiel interessant. Es ist ein Lehrstück guter Programmierung für jeden, der seinen Programmierstil verbessern möchte.

Bevor wir uns den programmiertechnischen Besonderheiten widmen, geben wir Ihnen eine kurze Anleitung zum Umgang mit der Magic-Clock. Je neun Uhren befinden sich auf der Vorder- und Rückseite. Die Aufgabe ist es, alle Uhrzeiten auf Zwölf zu stellen. Dazu dienen die Tasten + und —. Mit Plus erreichen Sie eine Bewegung im Uhrzeigersinn, mit Minus entgegengesetzt. Je nach dem Rad, an welchem Sie drehen, und der Stellung von erschiedenen Schaltern, lassen sich manche Uhren verstellen, andere hingegen nicht. Das Drehrad wechseln Sie durch die Eingabe eines Punktes gefolgt von einer Zahl von Eins bis Vier. Die Schalter werden allein betätigt durch die Eingabe einer Zahl von Eins bis Vier. Die Rückseite ist nicht unabhängig von der Vorderseite. Wenn etwa Schalter eins einen gewissen Schaltzustand zeigt, so zeigt Schalter zwei auf der Rückseite den entgegengesetzten. Wenn am Rad gedreht wird, drehen sich auch Uhren der Rückseite, aber nicht die gleichen wie auf der Vorderseite. Die Seiten vertauschen Sie durch die Eingabe der Zahl Null. Die Zahlen Eins bis Vier, der Punkt gefolgt von einer solchen Zahl, die Null und Plus- sowie Minuszeichen sind bereits alle Tasten zur Bedienungssteuerung. Ihr richtiger Einsatz sollte zum gewünschten Ergebnis führen.

Es gab bereits eine Rubik's Clock auf der **COMMODORE DISC** Nr. 29. Doch lief dieses Programm nur auf dem C128. Das jetzige Programm ist universellerer Natur, es ist nämlich nicht nur auf dem C128, sondern auch auf dem C64 und gar auf den Rechnern C116, C16 und dem Plus4 lauffähig. Auch wenn Sie einen C128 besitzen und mit Rubik's Clock von COMMODORE DISC Nr. 29 vollends zufrieden sind, könnte dennoch das Programm Magic- Clock von Interesse für Sie sein. Denn mein eigentliches Anliegen ist nicht, Ihnen lediglich ein Programm zum Gebrauch zur Verfügung zu stellen, sondern Sie vielmehr in guter Program- mierung zu unterweisen.

Schon wenn Sie im Directory die Programmlänge begutachten, fällt eines auf. Magic-Clock besitzt eine Programmlänge von 16 nur Blocks, während Rubik's Clock auf 76 Blocks kam. Eine spezielle C128-Version von Magic-Clock kam mit 14 Blocks aus. Nach Beseitigung der REM-Zeilen waren es gar nur mehr neun Blocks. Welch ein Unterschied besteht doch zwischen gutem und weniger gutem Programmierstil.

Wie schreiben Sie Ihre Programme? Haben Sie nur eine vage Idee und setzen sich dann gleich an den Rechnern, um mit der Codierung zu beginnen? Nichts ist bei einfachen Programmierproblemen gegen dieses Verfahren einzuwenden. Meist gehe auch ich in dieser Art und Weise vor. Doch wenn das Problem komplex und nicht leicht überschaubar ist, ist Drauflosprogrammieren nicht zu empfehlen. Außerdem zeichnet sich das Commodore-BASIC nicht gerade durch besonders gute Lesbarkeit aus. Während andere Programmiersprachen wie etwa Pascal mit der Möglichkeit, Unterprogrammen treffende Namen zu geben, brillieren, ist ein GOSUB auf eine Zeilennummer recht nichtssagend.

Auch läßt sich mit Variablennamen von maximal zwei Buchstaben nur recht unzureichend die eigentliche Aufgabe einer Variablen charakterisieren. Das sofortige Operieren mit programmiersprachlich en Syntaxen beschränkt des Menschen Schaffenskraft. Eingeengt in syntaktische Zwänge können Phantasie und Erfindergeist sich nicht völlig frei entfalten und müssen den Notwendigkeiten der Materie unnötigen Tribut zollen.

Daher erfolgte die Hauptarbeit der Programmierung weder in BASIC noch in Pascal. Als Sprache für den Entwurf nahm ich, was mir zur Verfügung stand. Es war dies keine spezielle Entwurfssprache, sondern die Programmiersprache FORTH, die sich auch hervorragend für Entwurfszwecke eignet. Sie selbst freilich brauchen sich nicht extra mit FORTH oder einer Ent- wurfssprache anfreunden; setzen Sie auf das, was Ideen Gestalt annehmen läßt, auf die Macht des Wortes. FORTH etwa erlaubt die Definition von Worten, die alles mögliche darstellen können, Operatoren, Funktionen, Unterroutinen, Prozeduren, Variablen und konstante

Werte und setzt dem, was durch das Wort bewirkt werden soll, keinerlei Grenzen. Allein dem Programmierer obliegt es, das vollbringende Wort zu definieren und auf die Ebene des vom Computer direkt Ausführbaren herabzutransformieren.
Mag das vielleicht noch etwas abstrakt, wenn nicht geradezu mystisch, geklungen haben, so werden wir gleich konkreter. Werfen Sie doch einmal einen Blick in das Programm Magic-Clock. Am Anfang nach der Rechneranpassung erfolgt in Zeile 130 der Sprung in die Zeile 1860. Was dort geschieht, erfahren Sie durch LIST 1850. Es ist die oberste Struktur des Programmes namens Hauptschleife. Wenn Sie die dort vorkommenden GOSUB-Befehle weiter durch Listen der um den Wert 10 kleineren Zeilennummer abfragen, erhalten Sie die Struktur, die in FORTH lautete:

ANFANG BEGIN EINGABE TASTWERTUNG AGAIN

Am Anfang müssen Variablen initialisiert werden. Außerdem ist die Magic-Clock auf dem Bildschirm darzustellen. Danach geht es in eine Endlosschleife, in der fortwährend die Tastatur abzufragen und die erfolgten Tastendrucke auszuwerten und weiterzuverarbeiten sind.
Sehr kurz ist diese Struktur, denn ein Programm soll vor allen Dingen überschaubar sein. Monsterroutinen über viele Zeilen zeugen nicht vom Drang nach Übersichtlichkeit. Empfehlenswert sind Unterroutinen in der Länge von maximal sieben Worten oder einer einzigen Bildschirmzeile. Die Eingaberoutine Zeile 1890 dürfte in etwa dieser Forderung gerecht werden. Die Unterroutine TASTWERTUNG ab Zeile 1800 dagegen ist etwas länger. Jedoch ist dies einerseits durch die Umsetzung einer INSTR-Anweisung mit Hilfe einer Schleife in das BASIC V2.0 des C64 und andererseits eine Sprungliste bedingt.
Einer Listenstruktur wollen wir auch eine gewisse Länge zubilligen. In unserem Falle wäre auch eine Verzweigung mit IF nicht ungeschickt gewesen, da hierbei nur fünf Fälle zu unterscheiden gewesen wären. Jedoch scheinen viele Hobby-Programmierer nur IF und THEN zu kennen, so daß sie sich die Methode mit INSTR und ON-GOTO auch einmal einprägen sollten.
In Zeile 1810 sind die Bedienelemente für die Magic-Clock aufgelistet. Sie wurden so gewählt, daß beim C128 lediglich auf den Zahlenblock der Tastatur zugegriffen werden braucht. Mit den Zahlen von eins bis vier werden die vier Schalter der Magic-Clock betätigt. Mit einem Punkt wird die Wahl des zu drehenden Rades eingeleitet. Die Zeichen Plus und Minus bewirken die Drehung des Rades im und gegen den Uhrzeigersinn. Mit einer Null kann die Magic-Clock gewendet werden, so daß die Rückseite zur Vorderseite wird. Die dazugehörigen Unterroutinen SCHALTUNG, RADWAHL, DREHUNG und WENDUNG wurden durch Substantive bezeichnet, zum Zeichen, daß dort noch Inneres und Äußeres, Auswertung der Eingabe, Verarbeitung und Ausgabe nebeneinander vorkommen.
Zum Rechnen eignen sich Buchstabenzeichen wenig. Zum Überführen der ASCII-Zeichen in der Eingabevariablen C$ in die für die Verarbeitung wichtigen Werte der Parametervariablen X dient die Routine NUMMER. Aus den Zahlzeichen von eins bis vier werden Zahlen von null bis drei. Auch vor sonstigen Buchstaben macht die Routine nicht halt. Wenn Sie bei RADWAHL nach dem Punkt ein A eingeben, würde die Routine keine Fehlermeldung ausgeben, sondern das A wie eine Eins behandeln. Die äußeren Werte sind um eins höher als die inneren, da der Mensch geneigt ist, eine Null für nichtig zu halten, während der Computer gerne bei null zu zählen beginnt. Für die Umwandlung von Plus und Minus bei der DREHUNG in die Zahlen 1 und -1 wurde nicht extra eine eigene Routine bemüht.
Der Verarbeitungsteil, das Innere also, besteht aus den Routinen SCHALTEN, DREHEN, WENDEN und RIEGELN. Für die RADWAHL genügt es, das gewählte Drehrad in der Variablen RA festzuhalten. Der Kernpunkt der ganzen Angelegenheit bildet das RIEGELN. Je nach Stellung der Schalter und Wahl des Drehrades sollen gewisse Uhren gesperrt oder entsperrt sein. Bei DREHEN werden die entsperrten Uhren im oder gegen den Uhrzeigersinn weitergedreht, die gesperrten Uhren dagegen ändern die Zeit nicht.
WENDEN tauscht die Objekte der Rückseite gegen die Objekte der Vorderseite. Den inneren Routinen obliegt die Betätigung der Objekte Schalter, Riegel und Uhren, die alle durch Variablen und gewisse Zahlenwerte repräsentiert werden. Den äußeren Routinen obliegt die Sichtbarmachung der durch die Zahl repräsentierten Objekte auf dem Bildschirm. Solche Routinen sind LOESCHRAD, ZEIGRAD, ZEIGSCHALTER, SCHALTBILD und ZEIGZEIT. Während ZEIGSCHALTER nur einen einzigen Schalter auf den Bildschirm bringt, zeigt SCHALTBILD alle vier Schalter. Die durch Substantive bezeichneten Routinen entflechten das Äußere vom Inneren, so daß wir uns daraufhin ganz auf die innere Logik konzentrieren können, ohne noch weiter an das Äußere denken zu müssen.

```
REM SCHALTUNG
    NUMMER
    SCHALTEN
    I=X
    ZEIGSCHALTER(I)
    RIEGELN
```

Nach NUMMER liegt die Nummer des Schalters für die innere Routine SCHALTEN als Parameter X vor. Es erfolgt daraufhin der Schaltvorgang. Der Schaltzustand des gewählten Schalters wird verändert. Diese Veränderung muß nachher durch ZEIGSCHALTER(I) auf dem Bildschirm angezeigt werden. Da sich die Schalterbetätigung auf die Drehbarkeit der Uhren auswirkt, muß durch RIEGELN eine entsprechende Verriegelung bewirkt werden.

```
REM RADWAHL
    LOESCHRAD(RA)
    EINGABE
    NUMMER
    RA=X
    ZEIGRAD(RA)
    RIEGELN
```

LOESCHRAD löscht das Drehrad auf dem Bildschirm, so daß der Benutzer sicher sein kann, auch die gewünschte Funktion gewählt zu haben. Nach EINGABE und NUMMER kann das neue Drehrad in RA festgehalten werden. Sodann wird es durch ZEIGRAD auf dem Bildschirm dargestellt. Zu guter Letzt muß noch die neue Verriegelung festgelegt werden.

```
REM DREHUNG
    X=1
    IF C$="-" THEN X=-1
    DREHEN
    ZEIGZEIT
```

Je nach Wahl von Plus oder Minus ist die Zeit zu erhöhen oder zu vermindern. Die Variable X erhält den zu addierenden Wert zugewiesen. Die nicht gesperrten Uhren sind zu DREHEN. ZEIGZEIT aktualisiert die Uhrzeiten auf dem Bildschirm.

```
REM WENDUNG
    LOESCHRAD(RA)
    I=RA:RA=FNEX(1)
    ZEIGRAD(RA)
    WENDEN
    SCHALTBILD
    ZEIGZEIT
```

Wenn wir den Finger am gewählten Drehrad haben und drehen die Uhr um, so stellen wir fest, daß das Drehrad die Position wechselt. Aus Rad Nummer null wird Rad Nummer eins oder aus Rad Nummer drei wird Nummer zwei. Das Drehrad wechselt seine Position auf die horizontal gegenüberliegende Seite. Mathematisch kann dies durch Verändern des untersten Bits ausgedrückt werden.
Hierzu dient die XOR-Funktion, die aber nur der C128 besitzt. Um dieses Manko zu beseitigen, wurde eine eigene Funktion FNEX(N) definiert. Da den benutzerdefinierten Funktionen nur ein Wertparameter mitgegeben werden kann, muß der zweite Parameter durch eine Variable übergeben werden. Dies ist bei FNEX(N) die Variable I, wie es aus der Definition der Funktion in Zeile 300 ersichtlich ist. Das Drehrad wird gelöscht und die neue Position berechnet. Durch ZEIGRAD kann es an der neuen Stelle dargestellt werden.
Die übrigen Objekte der Rückseite bekommen wir durch WENDEN auf die neue Vorderseite. Es müssen durch SCHALTBILD und ZEIGZEIT die bislang auf der Rückseite vorhandenen Objekte auf dem Bildschirm dargestellt werden. Riegeln ist nicht notwendig, da durch bloßes Umdrehen der Uhr sich nichts an der Verriegelung ändert. Die Riegel von Vorder- und Rückseite werden ja ebenso durch WENDEN ausgetauscht.
Die Rückseite könnte uns viel zu schaffen machen. Da nicht nur auf der Vorderseite etwas geschieht, sondern die Rückseite mit der Vorderseite stets in Beziehung steht, können wir sie bei der Programmierung nicht vernachlässigen. Es wäre denkbar, daß das Programm für beide Seiten doppelt so lang sein müßte als nur für eine Seite. Außerdem existieren ja keine getrennten Räder und Schalter für beide Seiten. Die Zugehörigkeit zu beiden Seiten scheint gegen getrennte Behandlung der Seiten zu sprechen.
Jedoch zeigen Beobachtungen, daß auf der Rückseite die gleichen Gesetze gelten wie auf der Vorderseite. Es läßt sich daher doch zur Vereinfachung eine Entkoppelung beider Seiten erreichen. Wir definieren uns einfach getrennte Schalter für Vorder- und Rückseite. Nur müssen wir diese zu Beginn so initialisieren, daß die Beziehungen stimmen. Wenn auf der Vorderseite alle Schalter eingedrückt sind, müssen sie auf der Rückseite alle herausgezogen sein. Wenn wir einen Vorderseitenschalter betätigen, müssen wir hinterher noch den entsprechenden, nämlich den horizontal gegenüberliegenden, Rückseitenschalter betätigen. Wenn auf der Vorderseite ein Rad im Uhrzeiger gedreht wird, ist hinterher das horizontal gegenüberliegende Rad auf der Rückseite gegen den Uhrzeigersinn zu drehen. Die Vorgänge SCHALTEN, DREHEN und RIEGELN können wir je in zwei Teilvorgänge zerlegen, etwa:

```
REM SCHALTEN
    TEILSCHALTEN
    WENDEN
    ENTSPRECHEND
    TEILSCHALTEN
    WENDEN
    ENTSPRECHEND
```

Nach dem Schaltvorgang auf der Vorderseite wird die Uhr gewendet. Auf der Rückseite darf nicht derselbe Schalter betätigt werden, sondern der, welcher dem Vorderseitenschalter entspricht. ENTSPRECHEND ist der Parameter X umzurechnen. Wir können jetzt auch die Rückseite TEILSCHALTEN und sorgen wieder für die Ausgangslage durch WENDEN und ENTSPRECHEND. Jedoch läßt sich der Code noch verkürzen, wenn WENDEN und ENTSPRECHEND gleich in die Routine TEILSCHALTEN übernommen wird, da dann diese Aufrufe nicht zweimal geschrieben werden brauchen.

```
REM SCHALTEN
    TEILSCHALTEN
    TEILSCHALTEN
```

```
REM DREHEN
    TEILDREHEN
    X=-X
    TEILDREHEN
```

Der Tausch der Drehrichtung ist nur einmal erforderlich, da wir diese hinterher nicht mehr benötigen.

```
REM RIEGELN
    X=RA
    TEILRIEGELN
    WENDEN
    I=X:X=FNEX(1)
    TEILRIEGELN
    WENDEN
```

Das Drehrad ist ein Faktor, der die Verriegelung bestimmt. Nach der Vorderseite kommt die Rückseite mit geänderter Drehradposition dran. Daß WENDEN in TEILRIEGELN integriert werden hätte können, habe ich offensichtlich übersehen. Nach der Entflechtung von Innerem und Äußerem, von Vorder- und Rückseite müssen wir uns, wenn wir es nicht bereits vorher taten, Gedanken über die inneren Objekte und die Form ihrer Verwaltung machen. Als Bestes erwies sich, alles in einer einzigen

dimensionierten Variablen unterzubringen. Beim WENDEN brauchen somit nicht alle Objekte auch tatsächlich ausgetauscht zu werden, sondern nur die Zeiger auf Vorder- und Rückseite.

```
REM WENDEN
    A=VO
    VO=HI
    HI=A
```

Ein Austausch der Variableninhalte VO (Vorne) und HI (Hinten) genügt. Weitere Variablen, die stets gleichbleibende Werte enthalten, ermöglichen den gezielten Zugriff auf die Objekte. SC ist der Offset für die Schalter, RG für die Riegel und ZE für die Zeit. Gespeichert sind alle Werte im Feld F(43). Das sind 44 Objekte, bestehend aus vier Schaltern, neun Zeiten und neun Riegeln, je für Vorder- und Rückseite. Die anfänglichen Vorderseitenobjekte befinden sich in F(0) bis F(21), die anfänglichen Rückseitenobjekte in F(22) bis F(43). Dementsprechend weist VO zu beginn den Wert null, HI den Wert 22 auf. Die Zeit der Uhr Nummer zwei erhalten wir durch F(2+ZE+VO). Für die anderen Objekte gilt analoges.

```
REM TEILSCHALTEN
    A=X+SC+VO
    I=F(A):F(A)=FNEX(1)
    WENDEN
    I=X:X=FNEX(1)
```

Aus dem Parameter X, dem Schalteroffset und dem Vorderseitenzeiger wird der Indexzeiger A berechnet. Mit der FNEX-Funktion wird der Schaltzustand geändert. Danach erfolgt die Vorbereitung für das TEILSCHALTEN der Rückseite.

```
REM TEILDREHEN
    AD=RG+VO
    FOR I=AD TO AD+8
    IF F(I)=1 THEN F(I+9)=FNKO(F(I+9)+X)
    NEXT
    WENDEN
```

AD ist der Zeiger auf den ersten Riegel. Die Zustände aller neun Riegel werden überprüft. Falls die Uhr entsperrt ist (F(I)=1), wird zur Zeit F(I+9) der Betrag X addiert und eine Korrektur FNKO vorgenommen, die die unzulässigen Zeiten null und 13 in die Werte zwölf und eins umsetzt.
WENDEN ist wieder die Vorbereitung für das TEILDREHEN der Rückseite.
Was noch für die Innere Logik fehlt, ist lediglich der Punkt TEILRIEGELN. Doch dieser dürfte wohl auch der schwierigste sein. Bei vier Schaltern und vier Rädern wären 64 verschiedene Sperrzustände denkbar. Doch keine Angst, so viele sind es nicht. Es herrschen glücklicherweise Symmetriegesetze, die die Möglichkeiten stark einschränken. Um aus den Schalterzuständen die einzelnen Fälle zu unterscheiden, gibt es verschiedene Wege. Der schlechteste, doch übliche ist eine Liste von IF-Anweisungen.

```
IF S0=0 AND S1=0 AND S2=0 AND S3=0 THEN FA=0
```

Statt sechzehn solcher Zeilen genügt auch eine einzige, wenn wir von binärer Rechnung Gebrauch machen:

```
FA=S0+2*S1+4*S2+8*S3
```

Dank der Symmetrieeigenschaften der Magic-Clock haben wir jedoch nur fünf Fälle zu unterscheiden. Wir zählen die gedrückten Schalter durch Addieren der Schaltzustände und erhalten Werte von null bis vier. Die dazugehörige Routine heißt EINSUMME(/N). Der Schrägstrich bedeutet, daß die Variablen rechts davon Ausgabeparameter sind. EINSUMME übergibt an N die Anzahl der gedrückten Schalter.
Ganz eindeutig ist die Sache noch nicht, denn bei zwei gedrückten Schaltern lassen sich noch zwei Möglichkeiten unterscheiden. Diese Schalter können entweder nebeneinander oder diagonal gegenüberliegend angeordnet sein. Der zweite Faktor ist das gewählte Rad. Wir brauchen hierbei auch nicht vier Fälle zu unterscheiden. Die Beobachtung zeigt nämlich, daß es nur darauf ankommt, welchen Schaltzustand der dem Drehrad benachbarte Schalter aufweist. Zehn verschiedene Fälle gibt es folglich insgesamt. Daran ändert auch der bei zwei gedrückten Schaltern hinzukommende Fall nichts, denn bei vier Schaltern unten kann der dem Drehrad benachbarte Schalter nicht oben sein. Das Umgekehrte gilt bei vier Schaltern oben.

```
REM TEILRIEGELN
    EINSUMME(/N)
    IF F(X+VO)=0 THEN OBFALL
    ON N GOTO
    EINKNOPFUNTEN,
    VERRIEGELN,
    VERRIEGELN,
    VERRIEGELN

REM OBFALL
    ON N GOTO
    ENTRIEGELN,
    ENTRIEGELN,
    ZWEIKNOPFOBEN,
    VERRIEGELN
```

Das meiste läßt sich bereits mit den zwei Standardroutinen ENTRIEGELN und VERRIEGELN lösen. Zwei Fälle erfordern die Spezialroutinen EINKNOPFUNTEN und ZWEIKNOFOBEN. Die ganze Logik hat sich erfreulicherweise ziemlich aufgelöst. Dies wurde durch die Beobachtung möglich, daß sich alle Eckuhren drehen lassen, deren benachbarter Schalter dieselbe Stellung aufweist wie der der Drehuhr benachbarte Schalter. In den meisten Fällen sind dann alle übrigen Uhren entweder gesperrt oder entsperrt.

```
REM VERRIEGELN
    W=0:ALLES(W)
    ECKRIEGELN

REM ENTRIEGELN
    W=1:ALLES(W)
    ECKRIEGELN
```

Alle Riegel werden entweder auf Sperren oder Entsperren gesetzt. Danach werden die Ecken gemäß der Schaltstellung geriegelt.

```
REM ECKRIEGELN
     FOR A=0 TO 3
     I=FNEI(A)
     I=FNEX(FNEI(X))
     F(VO+RG+EC(A))=FNEX(1)
     NEXT
```

Für alle vier Schalter wird der Schaltzustand FNEI(A) erfragt. Dieser wird durch exklusives Odern mit dem Schaltzustand FNEI(X) des dem Drehrad benachbarten Schalters vernüpft. Bei Gleichheit ergibt sich für I der Wert Null, bei Ungleichheit der Wert Eins. Das ist genau das Umgekehrte vom gewünschten Ergebnis, denn die Riegel werden mit Eins entsperrt. Doch mit FNEX(1) haben wir das Gewünschte.
Ein Problem besteht noch. Die Schalter sind numeriert von null bis drei, die Riegel dagegen von null bis acht. Doch ein weiteres Feld EC(3) ermöglicht uns, die Nummer der benachbarten Eckuhr oder des dazugehörigen Riegels zu finden.
Ein Spezialfall ist ZWEIKNOPFOBEN. Liegen sich die gedrückten Schalter diagonal gegenüber, ist ENTRIEGELN die richtige Lösung, liegen sie nebeneinander, ist zusätzlich die Uhr zwischen den den gedrückten Schaltern benachbarten Eckuhren zu sperren.

```
REM ZWEIKNOPFOBEN
     ENTRIEGELN
     I=X
     IF FNEI(X)=FNEI(FNEX(3))
     THEN WELCHE:F(RG+VO+A)=0
```

FNEI(X) ist der Zustand des dem Drehrad benachbarten Schalters. Nach I=X ist FNEI(FNEX(3)) der Zustand des diagonal gegenüberliegenden Schalters. Wenn die Schaltzustände ungleich sind, muß noch eine bestimmte Uhr gesperrt werden, aber welche? Die übergebene Variable A gibt darüber Bescheid.

```
REM WELCHE
     FOR I=0 TO 3
     IF FNEI(I)=1 THEN B=A:A=I
     NEXT
     A=FNZW(B)
```

Die vier Schalter werden abgefragt. Ist der Schalter gedrückt, wird die Nummer des Schalters in die Variable A übernommen. Vorher muß der frühere Inhalt von A nach B gerettet werden. Am Ende stehen in A und B die Nummern der zwei gedrückten Schalter zur Verfügung.
FNZW(B) stellt die Nummern der dazugehörigen Riegel fest, summiert und halbiert. Das Ergebnis, die Nummer der dazwischenliegenden Uhr, wird an A übergeben.
Die zweite Spezialroutine ist EINKNOPFUNTEN. Es drehen sich die Eckuhr am Drehrad und die drei Nachbaruhren. Wenn wir die Sache anders aufziehen, können wir, anstatt drei Uhren zu entsperren, auch zwei Uhren sperren.

```
REM EINKNOPFUNTEN
     ENTSPERREN
     I=X:I=FNEX(3):A=I
     F(VO+RG+FNZW(FNEX(1)))=0
     F(VO+RG+FNZW(FNEX(2)))=0
```

Nach I=X erhalten wir durch FNEX(3) die Ecknummer der diagonal gegenüberliegenden Uhr. Die horizontal und vertikal benachbarte Uhr ist zu sperren. Welche Nummern das sind, läßt sich mit der FNZW-Funktion feststellen. Der erste Parameter ist A, die Eckuhr, für welche wir beide Nachbarn suchen. Als Funktionsparameter nehmen wir FNEX(1) und FNEX(2). Das ist erstens die horizontal gegenüberliegende Eckuhr und zweitens die vertikal gegenüberliegende Eckuhr. Die jeweils dazwischenliegende Uhr ist durch den Wert Null zu sperren.
Damit ist es gelungen, das Programmierproblem der Magic-Clock zu lösen. Auf das Äußere, die Ausgaberoutinen, wurde weniger eingegangen, da hier ja nicht die eigentlichen Schwierigkeiten stecken.
Einige Besonderheiten seien am Rande erwähnt. Für BASIC 2.0 mußte auf die Cursorpositionierung mittels CHAR-Anweisung verzichtet werden. Cursorsteuerzeichen wären zwar eine Lösung gewesen, jedoch eine sehr langsame. Daher griff ich auf die PLOT-Routine, eine Kernalroutine mit dem Aufruf SYS 65520, zu.
Jedoch unterscheiden sich bei den verschiedenen Rechnertypen die Übergabeadressen für die CPU-Register. Durch Abfrage einer Speicherstelle, deren Inhalt sich bei den Rechnern unterscheidet, konnten die Rechnertypen und somit die Übergabeadressen RX und RY für die CPU-Register X und Y ermittelt werden.
Die Position der Magic-Clock auf dem Bildschirm ist leicht zu verändern. Es brauchen nur in Zeile 200 den Variablen YK und XK andere Werte zugewiesen werden. Beachten Sie aber dabei bitte, daß die Darstellung des Drehrades zusätzlichen Platz benötigt. Die Darstellung des Uhrrahmens, also äußerer Rahmen und Einrahmung der Uhrzeiten, erfolgt nicht mit einem fertigen Bild, sondern mit einer speziellen Rahmenroutine, die Sie unter Umständen auch in eigene Programme übernehmen können. Da das Programm recht änderungsfreundlich ist, dürfte es Ihnen kaum schwerfallen, vielleicht noch eine Anleitung einzubauen, die auf Tastendruck jederzeit aufgerufen werden kann. Die Routinen UHRRAHMEN, SCHALTBILD, ZEIGZEIT und ZEIGRAD erlauben Ihnen den erneuten Bildaufbau.
Wenn sich die originale Magic-Clock in irgendeinem Punkt anders verhalten sollte als hier im Programm, so bitte ich dies zu entschuldigen. Die Magic-Clock war ein Geschenk, das ich meinem Neffen verehrte. Die Programmierung erfolgte im nachhinein aus der Erinnerung heraus. Sollten Sie irgendeinen Unterschied entdecken, so dürfte eine Korrektur nach dieser ausführlichen Programmbeschreibung sicherlich keine Schwierigkeit mehr darstellen.
Ansonsten hoffe ich, Ihnen gute Programmierung vor Augen geführt zu haben. Sicherlich finden Sie auch noch einige Programmzeilen, die vielleicht Kopfzerbrechen bereiten. Versuchen Sie darüber nachzudenken. Bestimmt ergeben sich auch daraus weitere wertvolle Anregungen für Ihren Programmierstil.

am

League Manager Pro

König Fußball regiert

Alle Spiele einer Fußball-Saison können mit ihren Spielpaarungen und Ergebnissen von League Manager Pro verwaltet werden. Es versteht sich, daß eine Punkte- und Toretabelle Sie jederzeit über den aktuellen Tabellenstand informiert.

Mit League Manager Pro kann eine Fußball-Liga verwaltet werden, an der mindestens zehn, höchstes aber 22 Mannschaften teilnehmen. Eine neue Liga muß zunächst installiert werden. Danach können jederzeit Ligaspiele eingegeben werden. Jedes Spiel wird einzeln eingegeben, danach wird die Tabelle neu berechnet.
Im Hauptmenü läßt sich ein reverser Balken mit Cursor auf/ab über die Menüpunkte bewegen. Hat man einen Menüpunkt ausgewählt, muß mit Return bestätigt werden.

Install New League
Zuerst gibt man den Namen der Liga (etwa Bezirksliga oder 1. Bundesliga) und die Spielzeit (zum Beispiel 1989/90) ein. Nun bestimmt man mit Cursor links/rechts die Anzahl der Mannschaften (mit Return bestätigen) und gibt alle Mannschaften ein. Ist die Liga korrekt installiert worden, beantwortet man die Abfrage „Save installed Datas" mit ja, legt eine Datendiskette ins Laufwerk und startet den Speichervorgang mit Return (oder bricht mit Escape ab). Beantwortet man die Abfrage mit nein, kann man wählen zwischen: „Installierte Daten ändern" (=ja) oder „Menüpunkt abbrechen" (=nein).

Input League Game
Ein Ligaspiel kann nur dann eingegeben werden, wenn sich die entsprechende Liga im Speicher befindet. Zuerst muß man mit Cursor links/rechts bestimmen, an welchem Spieltag das Ligaspiel stattgefunden hat, und mit Return bestätigen.
Paarung: Nun müssen mit Cursor links/rechts erst die Heimmannschaft, dann die Gastmannschaft ausgewählt und jeweils mit Return bestätigt werden.
Resultat: Mit Cursor links/rechts bestimmt man die Anzahl der Treffer der Heimmannschaft und der Gastmannschaft (jeweils mit Return bestätigen).
Hat man das Ligaspiel korrekt eingegeben, beantwortet man die nachfolgende Abfrage mit ja, der Computer berechnet die neue Tabelle, und das Ligaspiel wird in die „League History", die beim Menüpunkt „Save Datas" als File gespeichert wird, übernommen.

Load or Save Datas
Liga laden: Entsprechende Datendiskette ins Laufwerk legen und Ladevorgang mit Return starten (oder mit Escape abbrechen). Sollte sich auf der Diskette keine Liga befinden, erscheint die Fehlermeldung „This Disk is NO Datadisk. Try again (Y/N)?"
Liga speichern: Entsprechende Datendiskette ins Laufwerk legen und den Speichervorgang mit Return starten (oder mit Escape abbrechen).
Wichtig: Sollte sich auf der Datendiskette bereits eine Liga befinden, wird diese überschrieben.

Show League Table
Show Table: Auf dem Bildschirm erscheint die aktuelle Ligatabelle. Besteht die Liga aus mehr als zwölf Mannschaften, kann man den ersten und den zweiten Teil der Tabelle abwechselnd auf den Bildschirm holen, indem man die Abfrage jeweils mit ja beantwortet. Mit nein gelangt man ins Hauptmenü zurück.
Reset League: Abfrage „Reset the League" mit ja beantworten, die Datendiskette mit der Liga, die sich im Speicher befindet, ins Laufwerk legen und den Resetvorgang mit Return starten oder mit Escape abbrechen. Es werden nun die Tabellenpositionen, Punkte und Tore aller Mannschaften auf Null zurückgesetzt. Die zurückgesetzte Tabelle wird wieder gespeichert und die auf der Diskette vorhandenen Spielpaarungen und Resultate der laufenden Saison werden gelöscht.

Show League Games
Games: Auf dem Bildschirm erscheint das erste Ligaspiel der Saison. Es gibt über spitze Klammern die Möglichkeit, sich alle Paarungen und Resultate der laufenden Saison anzusehen: > = ein Ligaspiel vor, < = ein Ligaspiel zurück.
Matchday: Dieser Menüpunkt bietet die Möglichkeit, sich alle Paarungen und Resultate eines Spieltages anzusehen. Dazu wählt man mit Cursor links/rechts den gewünschten Spieltag aus und bestätigt mit Return.
Team: Unter diesem Menüpunkt kann man sich alle Paarungen und Resultate einer Mannschaft ansehen. Dazu wählt man mit Cursor links/rechts die gewünschte Mannschaft aus und bestätigt mit Return.

Disk Drive Command
Catalog: Ausgabe des Inhaltsverzeichnisses einer Diskette auf dem Bildschirm. Gewünschte Diskette ins Laufwerk legen und Return drücken (oder mit Escape abbrechen).
Format: Zunächst muß man für die Diskette, die formatiert werden soll, einen Namen eingeben (oder ohne Eingabe Return drücken, um den Menüpunkt abzubrechen). Nun die entsprechende Diskette ins Laufwerk legen und den Formatiervorgang mit Return starten (oder mit Escape abbrechen). Nach dem Formatieren kann das Hauptprogramm (League Manager Pro) auf der Diskette abgespeichert werden. Dazu die Abfrage mit ja beantworten.
Scratch: Name des Files eingeben, der von der Diskette gelöscht werden soll (oder ohne Eingabe Return drücken, um den Menüpunkt abzubrechen). Nun die entsprechende Diskette ins Laufwerk legen und den Scratch-Vorgang mit Return starten (oder mit Escape abbrechen).

Finish the Program
Mit Escape kann das Programm verlassen werden, mit jeder anderen Taste gelangt man ins Hauptmenü zurück. Sollte sich noch eine Liga im Speicher befinden, wird gefragt, ob diese schon auf Diskette gesichert ist. Bestätigt man mit nein, erfolgt der Rücksprung ins Hauptmenü.

Tobias Thiessen

Hyprapaint

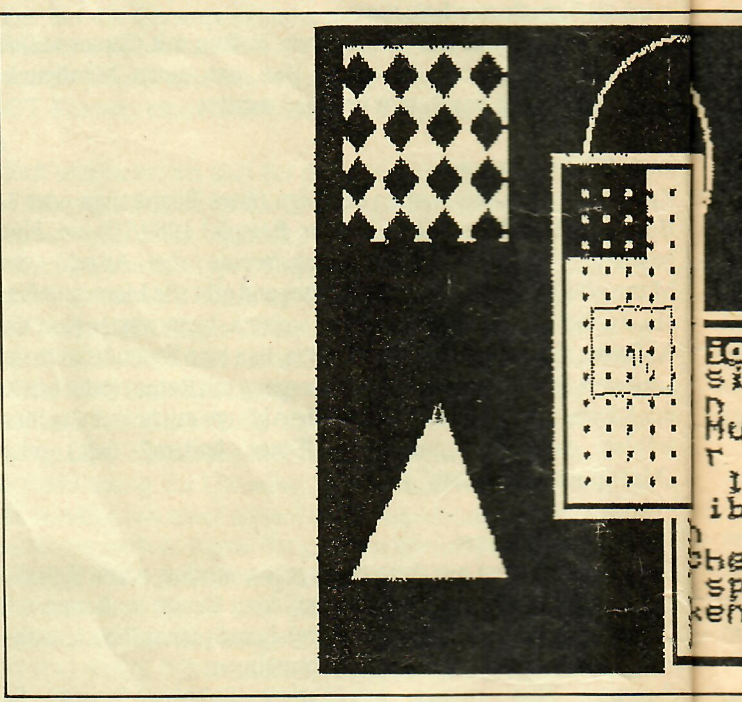
Malprogramm mit Pfiff

Eine ganze DIN-A4-Seite kann mit Hyprapaint bemalt werden. Umfangreiche Funktionen, komfortable Menüsteuerung und Hochgeschwindigkeit dank Maschinencode zeichnen das Programm aus.

Hyprapaint 128 ist ein Malprogramm, das komplett in Maschinensprache geschrieben ist und mit dem ein Hires-Bild mit einer Größe von 640 mal 800 Punkten (= eine DIN-A4-Seite) auf dem 40-Zeichen-Bildschirm erstellt werden kann. Gestartet wird das Programm auf dem Commodore 128 mit RUN "HYPRAPAINT 128".
Nach dem Start erscheint ein in zwei Teile geteilter Bildschirm. Auf dem linken Teil sieht man den momentan eingestellten Ausschnitt des Gesamtbildes, der 256 mal 200 Punkte groß und am Anfang noch leer ist. Auf ihm befindet sich ein blinkendes Kreuz, das mit einem Joystick in Port 2 oder einer Proportionalmaus (etwa Commodore 1351) in Port 1 gesteuert werden kann. (Gleichzeitiger Betrieb von Joystick und Maus ist möglich.)
Der rechte Bildschirmteil enthält eine Menüleiste, in der man unter den vier Befehlsgruppen Zeichnen, Flächen, Fenster und Sonstiges wählen kann. Man bewegt den Pfeil (das Kreuz verändert sich zu einem Pfeil, wenn man den linken Bildschirmteil verläßt) auf einen der vier Kästen und betätigt den Feuerknopf oder die linke Maustaste. Diese entspricht auch sonst dem Feuerknopf, während die rechte Maustaste das gleiche wie RUN/STOP bewirkt. Aus dem daraufhin erscheinenden Fenster kann man wieder eine Funktion auswählen, indem man den Pfeil auf die Funktion bewegt und Feuer drückt. Das Fenster verschwindet entweder nach Auswählen einer Funktion oder nach einem Abbruch, der durch Drücken von RUN/STOP beziehungsweise durch Drücken von Feuer außerhalb des Fensters bewirkt wird.
Unterhalb der vier Kästen in der Menüleiste wird in einem Feld die Farbe des Punktes, auf der sich das Kreuz befindet, angezeigt. In den nächsten beiden Zeilen sieht man, mit welcher Farbe auf einen schwarzen und mit welcher auf einen weißen Punkt gemalt werden soll. Diese Farben können durch Drücken von Feuer auf der entsprechenden Zeile der Menüleiste geändert werden. Darunter stehen x- und y-Koordinate des Kreuzes im Gesamtbild. Die unterste Zeile zeigt die augenblicklich angewählte Malfunktion an. Diese heißt nach dem Start „Linie".

Die einzelnen Funktionen

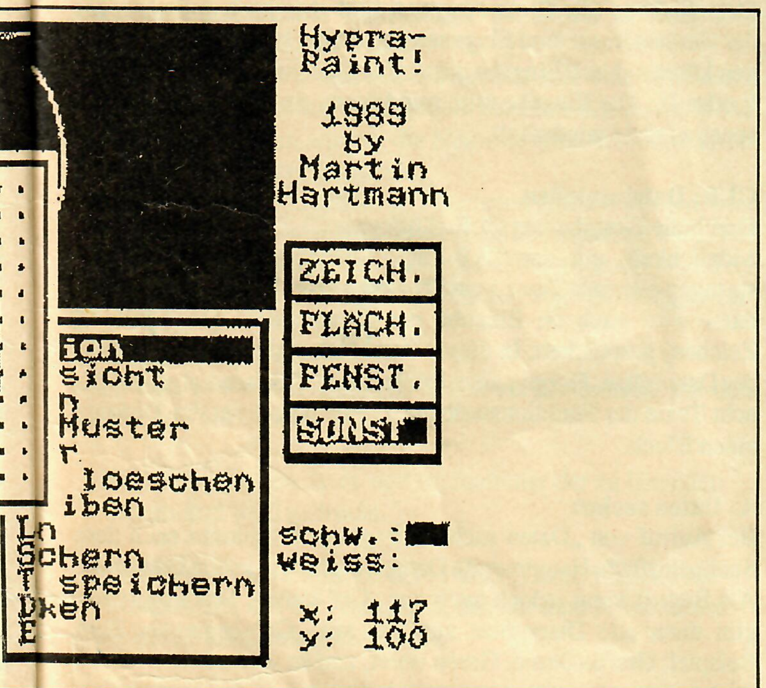
Die Funktion „Freihand" aus der ersten Befehlsgruppe ist eher für Besitzer einer Maus gedacht. Hält man Feuer gedrückt, wird an der Position des Kreuzes in der eingestellten Farbe ein Punkt gesetzt. Mit dem Joystick kann man auf diese Weise nur gerade und diagonale Linien ziehen. Die Funktion „Linie" bewirkt, daß hinter dem Kreuz nach einem ersten Druck auf Feuer eine Linie hinterhergezogen wird, die nach einem weiteren Druck von Feuer dann gemalt wird.
„Streckenzug" funktioniert wie Linie, nur daß nach dem zeichnen einer Linie an der Endposition eine neue Linie beginnt. Durch zweimaligen Druck von Feuer oder RUN/STOP kann der Anfang eines neuen Streckenzuges bestimmt werden. Die Funktion „Strahlen" bewirkt, daß alle Linien von einem anfangs zu bestimmenden Mittelpunkt ausgehen. Abgebrochen wird wie bei „Streckenzug". Nach Auswahl der Funktion „Quadrat" und Bestimmung des Anfangspunktes kann durch Bewegen des Kreuzes in x-Richtung ein Quadrat aufgezogen werden, dessen Höhe nachträglich durch eine Bewegung in y-Richtung geändert werden kann.
Mit „Rechteck" kann ein Rechteck durch Angabe von zwei Punkten aufgezogen werden. Die Funktion „Dreieck" erwartet das Anwählen von drei Punkten, die durch Linien verbunden werden.1 Bei „Parallelogramm" müssen wiederum drei Punkte eingegeben werden, der vierte wird vom Programm errechnet. Die Punkte werden ebenfalls durch Linien verbunden. Mit der



Mit Hyprapaint kann eine komplette DIN-A4-Seite gefüllt werden. Allerdings muß bei der Erstellung der Zeichnung in Ausschnitten gearbeitet we

Funktion „Kreis" kann nach Bestimmung des Mittelpunktes durch eine Bewegung in x-Richtung ein Kreis aufgezogen werden. „Ellipse" erwartet die Definition des Mittelpunktes und danach die x- und y-Radien, die durch eine Bewegung in die entsprechende Richtung bestimmt werden.
Sämtliche Funktionen der zweiten Befehlsgruppe werden in hoher Geschwindigkeit ausgeführt. „Quadrat", „Rechteck", „Dreieck", „Parallelogramm", „Kreis" und „Ellipse" funktionieren wie in der ersten Befehlsgruppe, nur werden die Körper mit der aktuellen Farbe beziehungsweise dem aktuellen Muster ausgefüllt. „Füllen" geschieht durch Bestimmen des Anfangspunktes. Eine Fläche, die

durch die andere Farbe begrenzt sein muß, wird in der aktuellen Zeichenfarbe beziehungsweise Muster gefüllt.
Die Funktion „Sprühen" erwartet zuerst die Bestimmung der Radien der Sprühfläche (siehe Ellipse). Hierzu wird der Zeichenbildschirm ausgeblendet. Danach werden bei gedrückter Feuertaste zufällig Punkte in aktueller Zeichenfarbe beziehungsweise Muster gesetzt. Durch Anwählen von „Füllmuster" kann das momentan eingestellte Muster für sämtliche Flächenfunktionen ein- und ausgeschaltet werden. Ist es eingeschaltet, wird hinter dem Namen „Füllmuster" ein Haken abgebildet. Ist es ausgeschaltet, werden die Flächen mit der eingestellten Zeichenfarbe gefüllt.
Die Funktion „Verschieben" in Befehlsgruppe drei erwartet zuerst die Definition eines Fensters (siehe „Rechteck"). Dieses kann an beliebiger Stelle des sichtbaren Bildausschnittes positioniert werden. Nach Drücken von RUN/STOP kann ein neues Fenster definiert werden. Um einen Bildausschnitt auf dem gesamten Bildschirm zu verschieben, kann nach der Definition des Ausschnittes Return betätigt werden. Ein spezielles Fenster wird eingeblendet, das wie „Position" zu bedienen ist (siehe unten). Bei „Größe ändern" muß zuerst wiederum ein Fenster definiert werden. Dessen x- und y-Größe kann jetzt beliebig verändert werden. Mit „Spiegeln (x/y)" wird das zu definierende Fenster an



den, da Hyprapaint nicht die komplette Seite auf einmal darstellen kann. Die Abbildung zeigt die komplette Seite und den nutzbaren Ausschnitt.

seiner x- beziehungsweise y-Mittelachse gespiegelt.
Mit „Fenster laden" kann ein bereits gespeichertes Fenster geladen werden. Beim Anwählen der Funktion sollte sich bereits eine Diskette im Laufwerk befinden. Alle auf der Diskette abgespeicherten Fenster werden auf dem Bildschirm angezeigt, und mit dem blinkenden Pfeil kann eines von ihnen ausgewählt werden. Auch diese Funktion kann mit RUN/STOP abgebrochen werden. Nach dem Laden verfährt man wie bei „Verschieben". Nach Anwählen von „Fenster speichern" muß zuerst ein Fenster definiert und dann ein Name eingegeben werden. Danach wird das Fenster gespeichert.

Die erste Funktion der vierten Befehlsgruppe ist „Position". In einem Fenster, das eine Verkleinerung des Gesamtbildes darstellt, zeigt ein inverses Rechteck, welcher Ausschnitt des Gesamtbildes im Moment zu sehen ist. Bewegt man den Pfeil in das Fenster hinein, wird an der Pfeilposition ein Rechteck angedeutet, das nach Drücken von Feuer als neue Position im Gesamtbild übernommen wird. Dies wird so oft wiederholt, bis außerhalb des Fensters Feuer gedrückt oder mit RUN/STOP abgebrochen wird. Mit „Übersicht" kann man das gesamte Bild in vierfacher Verkleinerung betrachten. „Zoomen" erlaubt eine achtfache Vergrößerung eines Bildschirmausschnittes, der mit Feuer ausgewählt wird. In dem Bildschirmausschnitt können wie bei der Funktion „Freihand" Punkte in der aktuellen Farbe gesetzt werden. Versucht man, den Pfeil aus dem Bildschirm hinauszubewegen, wird in die entsprechende Richtung gescrollt.

Kompatinilität ist alles

„Edit Muster" zeigt das aktuell eingestellte Muster in der rechten oberen Ecke an. Ein Quadrat kann beliebig auf dem Bildschirm bewegt werden, und durch Drücken von Feuer wird der Bildschirmausschnitt darunter als Muster übernommen. Mit dem Menüpunkt „Raster" kann ein Raster (alle 16 Pixel wird ein Punkt invertiert) ein- beziehungsweise ausgeschaltet werden. Ist es eingeschaltet, wird hinter dem Menüpunkt wie bei „Füllmuster" aus Befehlsgruppe zwei ein Haken abgebildet. Wenn die Funktionen „Sprühen" und „Freihand" aktiv sind, wird das Raster bei gedrückter Feuertaste kurzzeitig ausgeblendet.
Mit „Alles löschen" kann nach einer Sicherheitsabfrage das gesamte Bild gelöscht werden. Nach Wahl von „Schreiben" und der Definition des Startpunktes kann über die Tastatur ein Text eingegeben werden. Solange man nicht RUN/STOP oder Return betätigt hat, kann mit INST/DEL der letzte Buchstabe gelöscht und mit den Cursortasten die Textposition aktualisiert werden (der Text wird an die Pfeilposition verschoben). Mit Control schaltet man den Reversmodus an oder aus. Mit „Laden" kann ein abgespeichertes Gesamtbild geladen werden. Dabei verfährt man wie bei „Fenster laden".
„Speichern" erlaubt das Abspeichern des gesamten Bildes. Nach Eingabe eines Namens wird das Bild in gepackter Form (sonst würde es über 250 Blöcke einnehmen) auf Diskette gebracht.
„Teil speichern" ist dazu gedacht, gemalte Bilder in eigene Programme einzubinden. Bevor dieser Punkt angewählt wird, sollte die Position des Bildschirmausschnittes, der abgespeichert werden soll, festgelegt werden. Nach Eingabe des Namens wird ein 32 Blocks langes Files erzeugt. Um das Bild in eigenen Programmen anzuzeigen, genügt ein GRAPHIC 1,1: BLOAD "(name)". Mit „Drucken" wird nach einer Sicherheitsabfrage das gesamte Bild auf einem Epson-kompatiblen Drucker ausgedruckt. Mit „Ende" kann das Programm beendet werden.
Einige Tasten haben Sonderfunktionen: Space entspricht dem Feuerknopf. Mit INST/DEL kann der sichtbare Bildschirmausschnitt gelöscht werden. Die Taste U (undo) bewirkt, daß der Bildschirm, der bei der letzten Auswahl einer Funktion angezeigt wurde, wiederhergestellt wird. Die Commodore-Taste simuliert eine Dauerfeuerfunktion. Die Tasten 0 bis 9 bewegen den Pfeil an eine vorbestimmte Position. Zusammen mit Shift wird die aktuelle Pfeilposition gespeichert.

Martin Hartmann

File-Manager

Disketten verwalten

Ordnung in Ihre Diskettensammlung bringt File-Manager 128 V2.0. Er kann selbständig die Inhalte der Diskette lesen und sortieren, erlaubt aber auch manuelle Eingaben. Wenn Sie die so erstellten Listen ausdrucken, wissen Sie gleich, wo Ihre Programme zu finden sind.

Bevor Sie Daten erfassen können, ist dafür erst durch den Menüpunkt „Datei erstellen" eine Daten vorzubereiten. Da Sie auf der schreibgeschützten COMMODORE-DISC keine Daten abspeichern können, ist klar, daß Sie hierfür eine Datendiskette benötigen. Nach der Dateivorbereitung ist die Datei zwar vorhanden, aber noch nicht im Zugriff. Daher ist als nächstes der Menüpunkt „Datei öffnen" aufzurufen.

Die Befehlsübersicht

1 Hilfstext editieren
In dieses Untermenü gelangen Sie, wenn Sie im Hauptmenü die Taste 1 drücken. Wenn jedoch noch keine Datei geöffnet wurde, wird die Eingabe ignoriert. Im Untermenü wird die aktuelle Belegung der Funktionstasten angezeigt. Durch die Eingabe der Ziffern 1 bis 8 können Sie die Belegung ändern oder löschen, wobei beim Löschen mindestens ein alphanumerisches Zeichen eingegeben werden muß, etwa „-" oder „.", da sonst die Eingabe nicht abgeschlossen werden kann. Ins Hauptmenü gelangen Sie wieder durch die TAB-Taste.

2 Datei öffnen
In diesem Menüpunkt können Sie eine Datei öffnen, die vorher unter der Funktion „Datei erstellen" mit der CLR-Taste im Hauptmenü angewählt wurde. Das Programm will nun den Dateinamen (maximal 13 Zeichen) wissen. Jetzt lädt das Programm die verschiedenen Daten ein. Falls Sie sich im Hauptmenü vertippt haben sollten, können Sie durch Eingabe von *, anstelle des Dateinamens, wieder ins Hauptmenü gelangen.

3 Datei umbenennen
Die Funktion dient zum Umbenennen von Dateien auf der Diskette. Das Programm verlangt zuerst den alten und dann den neuen Namen (jeweils ohne Kennung eingeben). Man gelangt durch die Eingabe von * wieder ins Hauptmenü.
Achtung: Das Sternchen (*) wird nur bei der ersten Eingabe abgefragt!

4 Datei löschen
Diese Funktion dient zum Löschen von Dateien auf der Diskette. Hier muß auch die Kennung mit eingegeben werden. Man gelangt ebenfalls wieder durch den * ins Hauptmenü.

5 Directory
Hier wird das Directory ausgegeben. Die Ausgabe kann mit den üblichen Tasten (NO SCROLL, CBM) angehalten oder gebremst werden.

6 Diskette formatieren
Nach Eingabe des Diskettennamens und des Identifikationszeichens wird die eingelegte Diskette formatiert. Man gelangt durch Eingabe von * anstelle des Diskettennamens wieder ins Hauptmenü.

7 Parameter
Nach dem Aufruf von „Parameter" werden der aktuelle Dateiname (revers), die aktuelle Datensatzanzahl, der Floppyname, der Printername und die Geräteadresse der Datendiskette angezeigt. Falls noch keine Datei im Speicher ist, wird im Hauptmenü die Eingabe ignoriert. Voreingestellt sind Floppyname (Commodore 1571), Printername (Star LC-10C) und die Geräteadresse der Datendiskette (8). Diese Angaben sind aber auch änderbar, bei der Geräteadresse sind Eingaben von 8 bis 11 möglich. Änderungen können durch Eingabe des Anfangsbuchstabens erfolgen (F, P oder D). Die Eingabe muß mindestens ein Zeichen lang sein, sonst wird sie wiederholt.

CLR: Datei erstellen
Durch die Eingabe von CLR (Shift plus CLR Home) besteht die Möglichkeit, mit der TAB-Taste wieder ins Hauptmenü zu springen oder mit Taste 1 eine Datei zu erstellen. Die Parameterdatei wird nach de Eingabe des Dateinamens (maximal 13 Zeichen) gespeichert. In dieser Datei stehen die aktuelle Anzahl der Datensätze, Floppyname, Printername, Geräteadresse und die acht Texte der Funktionstasten. Das File belegt auf der Diskette einen Block.

A: Daten suchen
Bei Aufruf von „Daten suchen" fragt das Programm nach dem Suchbegriff (*=Hauptmenü). Danach sucht es in jedem Datensatz. Als Begriff kann jedoch auch nur „Gr" stehen. Das Programm gibt dann alle Datensätze aus, die mit „Gr" beginnen. Zum Beispiel: Grafik-Demo, Green Beret, und so weiter.
Achtung: Wenn Sie etwa „1.1" als Diskettennummer eingeben, werden auch die Datensätze mit der Nummer „1.11", „1.12" et cetera ausgegeben.

B: Daten eingeben
Nach dem Aufruf im Hauptmenü erscheint ein weiteres Menü mit den folgenden Funktionen:

1 — Daten eingeben

2 — Daten einlesen

TAB — Hauptmenü

Durch die TAB-Taste gelangen Sie wieder ins Hauptmenü. Durch

die Ziffemtaste 1 können Sie die Daten selbst eintippen. Dazu erscheint die aktuelle Datensatznummer. Danach können Sie den Filenamen (*=Untermenü), die Programmart, die Diskettennummer und die Boxnummer, falls Sie mehrere Diskettenboxen besitzen, eingeben. Sobald Sie alle Eingaben mit der Return-Taste abgeschlossen haben, wird der Datensatz auf Diskette geschrieben.
Achtung: Das Programm ist auf die Eingabe von mindestens einem Zeichen eingestellt, damit es keine Probleme mit der Datenspeicherung gibt.
Nach Betätigung der Taste 2, werden Sie aufgefordert, die zu lesende Diskette einzulegen. Mit der TAB-Taste können Sie die Funktion abbrechen, die Return-Taste setzt das Programm fort. Nun wird das Diskettenformat (1541) oder 1571) angezeigt. Falls es sich um eine Diskette im 1541er Format handelt, muß die Rückseite separat gelesen werden. Danach wird das Disketten-ID angezeigt, das auch als Diskettennummer gilt. Jetzt werden Sie aufgefordert, die Boxnummer (A-F) einzugeben. Sogleich beginnt das Programm die Filenamen einzulesen. Als Programmart wird der Bindestrich (-) angegeben.
Achtung: Trennlinien im Directory (——————) werden nicht mit eingelesen! Wenn alle Namen gelesen sind, erden die Daten alphabetisch sortiert und in Namen mit Großbuchstaben geändert. Diese Arbeit kann einige Sekunden in Anspruch nehmen, also ein bißchen Geduld.
Im nächsten Teil werden alle Filenamen mit einer Nummer versehen und zweispaltig ausgegeben. Sie können nun, wenn Sie einen Namen löschen wollen, die Nummer eingeben oder durch die Zahl 99 die Daten speichern.
Achtung: Es können maximal 30 Filenamen gelesen werden. Das Programm springt weiter, wenn diese Zahl überschritten wird. Mit der TAB-Taste gelangen Sie ins Untermenü, ohne die Daten zu speichern. Durch die Return-Taste wird das Programm fortgesetzt.

G: Daten listen (PRT)
Nach Drücken von G gelangen Sie in ein Untermenü mit den folgenden Funktionen:

1 Nach Eingabe von Start- und Endnummer der zu listenden Datensätze wird gedruckt.

2 Diese Funktion ist ähnlich dem Punkt „Daten suchen" im Hauptmenü, mit der Ausnahme, daß auf dem Printer anstelle des Bildschirms gelistet wird.

Mit der TAB-Taste gelangen Sie ins Hauptmenü.
Die Druckeransteuerung gilt für den Star LC-10C. Man kann und muß sie, falls Probleme auftauchen, im Programm ändern.
Achtung: Gelöschte Datensätze werden nicht mit abgedruckt!

ESC: Programm beenden
Hier gelangen Sie wieder in ein Untermenü, das folgende Funktionen aufweist:
SPACE — Daten speichern. Es wird die alte Parameterdatei gelöscht und die aktuelle gespeichert. Danach wird das Programm neu gestartet.
ESC — Programm beenden (END). Der END-Befehl wird ausgeführt. Neustart ohne Datenverlust mit "GOTO 25".
Mit der TAB-Taste gelangen Sie ins Hauptmenü.
Fehlermeldungen
Floppy- & BASIC-Fehler werden in einem Fenster eingeblendet. Nach der Bestätigung durch die RETURN-Taste gelangen Sie wieder ins Hauptmenü.
Achtung: Falls kein Laufwerk oder Drucker angeschlossen oder eingeschaltet ist, wird das Programm abgebrochen.
Tip: Wenn es sich um den Diskettenfehler 70, NO CHANNEL, 00, 00 handelt, sollte die Floppy aus- und wieder eingeschaltet werden.

Pascal Künzli

DIALOG

Ladeproblem beseitigen

Für schlecht ladende Disketten im C128-Modus, die entweder einseitig bespielt sind, oder beidseitig, wobei jede Diskettenseite für sich einseitig formatiert wurde, gibt es eine recht einfache Lösung

Die Floppy 1571 läßt sich mit einem Befehl so umstellen, daß eine Floppy 1541 daraus wird, die dieses Rückseitenproblem nicht kennt. Stecken Sie erst nach dem Einschalten oder nach einem Reset, wenn die Meldung READY erschienen ist, die Diskette in das Laufwerk und geben Sie ein:

```
OPEN8,8,15,"U0>M0":
CLOSE8
```

Nach dem Abschicken dieser Zeile mit Return ist die Diskette einwandfrei lesbar bis zum nächsten Reset, wenn außer diesem Rückseitenproblem kein Diskettenfehler vorliegt. Sollten Sie einmal die Rückseite einer nur einseitigen bespielten Diskette beschrieben haben, sollte die COMMODORE DISC im C128-Modus Ladeschwierigkeiten haben: Mit dieser Zeile können Sie dies beheben. Wenn Sie bei der COMMODORE-DISC danach den BOOT-Befehl eingeben, sollte gar ohne jegliches Laufwerkflackern das Auswahlmenü erscheinen.

Sortierproblem

Hogenacker aus Overath hatte eine Idee für ein Spiel. Das Problem war: Fünf Zahlenwerte mußten dabei sortiert werden. Jedoch funktionierte seine Sortierroutine nicht richtig, war außerdem recht umständlich und fast unüberschaubar.
Hier ist unser Rat: Das Sortieren ist keine allzu komplizierte Geschichte. Mit drei Programmzeilen läßt sich das Problem lösen:

```
FOR J=1 TO D-1:
FOR I=1 TO D-J
IF A%(I) > A%(I+1)
THEN
A=A%(I):A%(I)=A%(I+1):A
%(I+1)=A
NEXT I: NEXT J
```

Sortiert wird hierbei das Feld A%(D). Bei 20 Werten ist der Algorithmus bereits ziemlich langsam. Für fünf Werte dürfte er aber einer der Schnellsten sein. Durch seine Kürze und Einprägsamkeit ist er wohl die optimale Lösung für Ihr Problem.

Fehler in BIGBOSS

Günter Kramer aus Emsdetten entdeckte noch einen Fehler in seinem Programm „Bigboss" auf der COMMODORE DISC 27. Eine Sprungadresse ist zu ändern:

```
alt:10820
IFE<>32THEN10820
neu: 10820
IFE<>32THEN10800
```

Wenn die Zeile nicht geändert wird, hängt sich das Programm an dieser Stelle in einer Endlosschleife auf.

Sortierverfahren im Vergleich

Gut sortiert ist halb gefunden

Daten in den Computer einzutippen, sie wieder auf dem Bildschirm oder Drucker auszugeben, ist für die meisten ein Kinderspiel. Schwieriger wird es allerdings, wenn es darum geht, Ordnung in die Daten zu bringen, sie zu sortieren.

Irgend ein Schlaumeier hat einmal das geflügelte Wort von sich gegeben: „Die Hälfte seines Lebens sucht der Mensch vergebens...". So banal der Spruch auch klingt, ein Körnchen Wahrheit steckt doch drin.
Jede Kartei, jedes Verzeichnis richtet sich nach gewissen Suchkriterien aus. Der gewünschte Name, die gesuchte Adresse soll ja ohne Zeitverlust möglichst nervenschonend gefunden werden. Bei der Dateiablage im Computer ist der Idealfall, die geforderten Daten auf Knopfdruck abrufen zu können. Sicherlich findet sich der Computer auch in ungeordneten Datenbeständen zurecht, es dauert nur länger. Wenn aber eine Liste ausgegeben ist, aus der der Mensch etwas erkennen soll, so findet er sich bei größerer Länge kaum mehr zurecht, wenn nicht eine Sortierung das Auffinden erleichtert.
Es gibt verschiedene Möglichkeiten, zum Ziel zu kommen. Da wir ziemlich sicher sind, daß der größte Teil unserer Leser in BASIC programmiert, wollen wir Ihnen hier ein paar Sortier-Unterprogramme vorstellen. Die Bezeichnung „Unterprogramme" sagt es eigentlich schon: Sie sind dazu gedacht, irgendwo in eine Ihrer eigenen Software-Entwicklungen eingebaut zu werden.
Wir unterscheiden verschiedene Arten von Sortieralgorithmen. Es gibt Sortieralgorithmen mit begrenzten Anwendungsmöglichkeiten und Sortieralgorithmen für alle Fälle. Bei letzteren lassen sich einfache Algorithmen unterscheiden, die bei geringer Datenmenge sehr schnell sind, jedoch bei größeren Datenmengen äußerst langsam werden. Bei komplexeren Algorithmen wächst die Sortierzeit nicht überproportional zur Datenmenge.

Hash-Sort

Ein einfacher und äußerst schneller Sortieralgorithmus für begrenzten Wertebereich ist der Hash-Sort. Zum vollständigen Sortieren genügt ein einziger Durchlauf. Ein spezieller Anwendungsfall wäre etwa eine Lottoauswertung für Systemspieler. Wieviel Dreier, Vierer, Fünfer, Fünfer mit Zusatzzahl und Sechser sind innerhalb eines Jahres erziehlt worden? Daß das Sortieren so schnell geht, liegt daran, daß jedem vorkommenden Wert ein Speicherplatz zugeordnet ist, in dem die Anzahl des Vorkommens erfaßt wird. Aus dem zu untersuchenden Wert muß dieser Speicherplatz, ohne noch zu suchen, errechenbar sein.
Wenn der Wertebereich zu groß ist, etwa Zahlen zwischen 0 und 100000 enthält, ist nicht genügend Speicherplatz vorhanden. Wenn der Wertebereich ein Vielfaches der Datenmenge beträgt, geht das Durcharbeiten des Datenbestandes zwar auch verflixt schnell, das Auslesen des Vorkommens dagegen braucht dann doch seine Zeit. Zahlen sind für den Hash-Code kein Problem. Für die Sortierung von Wörtern muß der Hash-Algorithmus erheblich ausgebaut werden. Vielleicht sollte dann aber lieber ein anderer Algorithmus gewählt werden.
Einfache Algorithmen sind die schnellsten, wenn nur einige Werte zu sortieren sind. Das liegt einmal daran, daß sie einfach gebaut sind und die Abarbeitung des Programmcodes deshalb nicht viel Zeit kostet. Die Langsamkeit bei größeren Datenmengen ist darauf zurückzuführen, daß die Anzahl der benötigten Vergleiche mit dem Quadrat der Datenmenge wächst. Für die zehnfache Datenmenge wird die hundertfache Zeit nötig.

Select-Sort

Relativ einfach ist der Select-Sort. Es sind so viele Durchgänge nötig, wie die Datei Elemente besitzt. Bei jedem Durchgang wird das größte Element herausgesucht. Oder wir suchen nach dem kleinsten, was genauso möglich ist. Am Ende des Durchganges steht das gesuchte Element fest. Das zuerst am Dateiende stehende Element wird gegen das jeweils größte ausgetauscht, oder das kleinste mit dem am Anfang stehenden.
Lassen wir letzteren Fall aber ruhig beiseite, denn wer die Suche nach dem größten Element verstanden hat, wird auch das Umgekehrte vollbringen können. Das größte Element steht nach dem ersten Durchgang auf dem ihm gebührenden Platz. Als nächstes ist das zweitgrößte zu finden und auf den zweithöchsten Platz abzulegen. So geht es weiter, bis die ganze Datei geordnet ist. Die Summierung der Durchgänge, multipliziert mit den jeweils nötigen Vergleichen, führt zur Formel:

ZV=N*(N-1)/2

ZV ist dabei die Gesamtzahl der Vergleiche, N die Anzahl der Daten. Dies bedeutet quadratisches Wachstum.

Bubble-Sort

Dasselbe wie der Select-Sort kann auch der Bubble-Sort bewirken, wenn wir mit ihm auf ähnliche Weise verfahren. Beim Select-Sort übernahmen wir den jeweils höheren Wert in eine Variable. Beim Bubble-Sort werden zwei hintereinander vorkommende Elemente verglichen und gegebenenfalls ausgetauscht. Nachteilig ist, daß dieser Austausch mehr Zeit kostet, als lediglich die Übernahme in eine Variable.
Doch gibt es die Möglichkeit, einen intelligenteren Bubble-Sort zu entwickeln. Nehmen wir einmal den Fall an, wir würden immer nach dem kleinsten Element suchen. Jedesmal wäre die Datei von hinten bis zum neuen Anfang zu durchsuchen. Der Datenvertausch erlaubt uns auch ein umgekehrtes Vorgehen. Anstatt von hinten nach vorne zu gehen, können wir beim jeweils nächsten Element beginnen und den Bubble bis zum Datenanfang laufen lassen. Dies bewirkt den Aufbau einer Teildatei, in die das nächste Element immer richtig einsortiert wird.
Das geht aber auch nicht schneller als der umgekehrte Weg, wenn wir nicht noch eine besondere Vorkehrung einbauen. In dem

Augenblick, da kein Austausch mehr nötig ist, hat unser Element den ihm zukommenden Platz gefunden. Ein weiteres Vergleichen bis zum Anfang ist damit unnötig und wäre reine Zeitverschwendung. Dieser Bubble-Sort, der sich auf der Disk befindet, dürfte bei ungeordneten Mengen kaum langsamer als der Select-Sort sein. Bei bereits geordneten oder teilweise geordneten Mengen ist er jedenfalls schneller. Nur ein einziger Durchgang wird bei bereits bestehender Ordnung durchgeführt. Bei einem einzeln auftauchenden ungeordneten Element würden sich etwa zwei Durchgänge ergeben. Bubble-Sort eignet sich daher auch nicht schlecht zur Einsortierung einzelner neuer Elemente.

Insert-Sort

Das Vertauschen von zwei Elementen bei jedem Vergleich kostet Zeit, die gespart werden könnte. Im Prinzip aufgebaut wie der Bubble-Sort, spart sich Insert-Sort diesen Tausch. Das einzusortierende Element wird, ähnlich wie beim Selekt-Sort, in einer Variablen gehalten. Das als zu groß befundene Element wird aus der Datenliste um einen Platz nach hinten versetzt. Ist das einzusortierende Element größer, kann es einfach eingesetzt und der jeweilige Durchlauf beendet werden. Insert-Sort ist in jedem Fall etwas schneller als Bubble-Sort. Bei ungeordneten Mengen dürfte er ein klein wenig schneller als Select-Sort sein.

Quick-Sort

Die einfachen Algorithmen besitzen den Nachteil, jedes Element fast mit jedem anderen zu vergleichen. Doch dies ist bei größeren Datenmengen zeitraubend. Auch mit weitaus weniger Vergleichen kann das Ziel erreicht werden. Das bewirkt die Halbierungsmethode. Davon macht Quicksort Gebrauch. Wenn wir eine Datei vorsortieren, so daß in der einen Hälfte alle Elemente größer sind als in der anderen, braucht der eine Teil nicht mehr mit dem anderen verglichen zu werden.
Zwar kann Quick-Sort nicht ein Vergleichselement so auswählen, daß die Datei genau halbiert werden kann, aber die Wahl des sich in der jeweiligen Mitte befindlichen Elementes genügt, um gute Annäherung an das Ideal zu erreichen. Quick-Sort führt diese Teilung so lange aus, bis jeweils ein einziges Element übrigbleibt, das damit an der richtigen Stelle steht. Quick-Sort ist bei kleinen Datenmengen langsamer als die einfacheren Algorithmen. Ab etwa vierzig Daten dürfte es vorne liegen.

Turbo-Sort

Turbo-Sort ist ein Versuch meinerseits, den Quick-Sort schneller zu machen. Weil bei kleineren Datenmengen die Sortierung langsamer war, gedachte ich, zwei Sortierverfahren zu mischen. Bereits bei etwa 18 Datensätzen hat Turbo-Sort die Nase vorne. Turbo-Sort ist bei kleineren Datenmengen nur unwesentlich langsamer als der Insert-Sort, bei größeren Datenmengen schneller als Quick-Sort. Halbiert wird die Datei nicht, bis nur noch ein Element übrig bleibt, sondern bis es weniger als dreizehn sind. Den Rest übernimmt eine Insert-Routine.
Bei bereits geordneten Mengen schneiden Bubble- und Insert-Sort am besten ab. Select-Sort und Quick-Sort nehmen eine komplette Neusortierung ohne Rücksicht auf eine bereits vorliegende Ordnung vor. Turbo-Sort verzichtet zumindest auf die Sortierung des der Insert-Routine verbleibenden Restes. Alle Algorithmen sind in zwei Versionen vorhanden: für Integerzahlen und Textstrings. Außerdem können Sie bei den Integer-Sorts eine beliebig große Datenmenge von Zufallszahlen zu Testzwecken auswählen. Eine Zahl als Vielfaches von sechzigstel Sekunden gibt Aufschluß über die Sortierzeit. Damit finden Sie sicherlich den für Ihre Zwecke optimalen Algorithmus.

am

Reuma

Frisch gemahnt...

Nicht jeder ist Geschäftsmann. Doch viele Computerbesitzer bieten selbstentwickelte Programme und ähnliches über Annoncen in diversen Zeitschriften an. Den Schriftverkehr kann der Computer wesentlich erleichtern.

Nachdem man das Programm geladen und gestartet hat, muß zuerst das Datum eingegeben werden (tt.mm.jj oder tt.mm.jjjj). Danach gelangt man ins Hauptmenü. Dort wird unter dem Informationskasten die zur Zeit bearbeitete Rechnung angezeigt (nach dem Start natürlich noch keine). Darunter steht das Datum und nachfolgend die fünf Menüpunkte: Adresse, Rechnung, Mahnung, Drucken und Ende.
Jeden Menüpunkt erreicht man durch Eingabe der jeweiligen Ziffer.

1 Adresse
Hier kann man die Adresse des Empfängers der Rechnung/ Mahnung eingeben. Zur Verfügung stehen:

Name 1,
Name 2,
Straße,
PLZ/Ort.

Bei *Name 1* kann man zum Beispiel Herrn, Frau, Fräulein oder Firma eintragen. Aber auch für Doppelnamen kann man *Name 1* und *Name 2* verwenden. Zum Schluß läßt sich noch das Lieferdatum der Ware notieren.

2 Rechnung
In diesem Menüpunkt muß zunächst die Nummer der zu schreibenden Rechnung eingegeben werden. Danach kann sofort mit dem Schreiben begonnen werden. Zuerst wird die Artikelnummer des Artikels 1 eingegeben, dann seine Bezeichnung und anschließend der Einzelpreis. Vor Pfennigbeträgen wird ein Punkt (.) und nicht ein Komma (,) gesetzt. Bei Preisen ohne Pfennige kann man die Nachkommastellen weglassen. Jede Eingabe wird mit Return abgeschlossen.

Nachdem man eine Zeile geschrieben hat, hat man folgende Wahlmöglichkeiten:

- Weiter = nächsten Artikel eingeben
- Ende = alle Artikel sind eingegeben
- Korrektur = letzten Artikel korrigieren

Bei „Korrektur" erscheint im Eingabefenster die Nummer des letzten Artikels noch einmal, und man kann ihn neu eingeben. Man braucht übrigens keine Angst zu haben, daß alles nach oben wegscrollt, wenn der Bildschirm voll ist. Er wird frühzeitig gelöscht, und man kann oben weiterschreiben.

Programmstart ins Listing getippt werden. Dies geschieht in den Zeilen 750 bis 870. In Zeile 750 und Zeile 770 wird der Text der Optik wegen breit gedruckt. Bei der Eingabe des Briefkopfes sollte man deshalb darauf achten, daß der Text nicht zu lang ist, da er sonst nicht mehr in die Zeile paßt. Wenn man keinen breiten Briefkopf haben möchte, muß in Zeile 750 und 770 das chr$(14) am Anfang gelöscht werden. Man hat dann auch mehr Platz für Text im Kopf.
Da in Zeile 750 und 770 der Firmenname steht, in den Zeilen 800, 810, 820, 850, 860 und 870 Adresse, Telefon-Nummer, Bankverbindung und Konto-Nummer eingegeben werden (die 55 nach den Anführungszeichen muß stehenbleiben, da diese Zahl

Hier Briefkopf eingeben
Hier auch Briefkopf eingeben

Straße...
PLZ Ort...
Tel.: 00000/00000

Firma
Hans Hansen
Hansweg 3

0815 Hanshausen

Bank...
BLZ...
Kto.-Nr. 0000 000

Datum: 22.7.1989

Rechnung Nr.0000 Lieferung vom 12.12.88

Stück	Artikel	E-Preis	G-Preis
12	Commodore 128	800.00	9600.00
12	Floppy 1571	500.00	6000.00
12	Monitor Commodore 1901	500.00	6000.00
12	Disketten (10'er Pack)	24.95	299.40

Nettobetrag	21899.40
+ 14% Mwst	3065.92
Gesamtbetrag	24965.32

Zahlbar: 10 Tage / 4% Skonto oder 30 Tage netto Kasse

Die Ware bleibt bis zur völligen Bezahlung mein Eigentum.

3 Mahnung
Da es öfter vorkommt, daß Rechnungen nicht sofort bezahlt werden, kann man mit Reuma natürlich auch Mahnungen schreiben.
Manchmal müssen bei einer Person sogar mehrere Rechnungen angemahnt werden, daher muß zuerst die Anzahl der anzumahnenden Rechnungen eingegeben werden.
Dann werden die Rechnungsnummer, das Rechnungsdatum, das Fälligkeitsdatum und der Gesamtbetrag der Rechnung eingegeben. Nach einer Sicherheitsabfrage geht es mit der nächsten Rechnung weiter. Zum Schluß muß die Mahnstufe eingegeben werden, dann geht es wieder ins Hauptmenü.

4 Drucken
Da man über diesen Menüpunkt Rechnungen und Mahnungen drucken kann, muß man zuerst eingeben, was gedruckt werden soll (r für Rechnung, m für Mahnung). Danach muß der Drucker angeschaltet werden, und nach Return wird gedruckt.
Da es eine sehr umständliche Sache wäre, wenn man vor jedem Drucken einen Briefkopf eingeben müßte, muß er schon vor dem

die Druckspalte angibt). Das Programm druckt jede Rechnung/Mahnung zweimal aus, einmal zum Verschicken und einmal für die Akten zu Hause. Wird dieser doppelte Ausdruck nicht gewünscht, muß man in Zeile 720 die Zwei in eine Eins umwandeln. In Zeile 1160 und Zeile 1480 müssen die Zahlungskonditionen geändert werden, falls sie nicht stimmen.
Nachdem man sein Programm an seine persönlichen Ansprüche angepaßt hat, sollte man es noch einmal abspeichern, um diese Änderungen nicht immer wieder vornehmen zu müssen.
Das Programm macht übrigens beim Ausdruck von mehr als 20 Artikeln einen Übertrag auf die nächste Seite und druckt dort weiter. Mehr als 50 Artikel können aber dennoch nicht gedruckt werden. Darauf ist auch schon bei der Artikeleingabe zu achten.
Nach dem Ausdruck gelangt man wieder ins Hauptmenü.
Der Punkt „Ende" beendet das Programm.

Reuma Programm läuft im DIN-Zeichensatz. Außer dem C128 wird folgende Hardware benötigt: Floppy 1541/70/71, 80-Zeichen-Monitor, Commodore-MPS-801-Drucker.

André Marcour

Landstraße

Vorsicht, Schlaglöcher

Daß mit wenig Aufwand bereits ein Action-Game, wenn auch kein überaus großartiges, geschrieben werden kann, zeigt das Programm Landstraße. Es geht darum, sich mit einem Fahrzeug über eine Landstraße zu bewegen. Das wäre keinerlei Problem, wenn es keine Schlaglöcher gäbe. Doch diese bilden sich abrupt und vermehren sich ständig. Die Schlaglöcher hemmen nicht nur die Bewegung des Autos, sondern kosten auch wertvolle Energie. Dadurch können Sie sich abhängig von der Anzahl der durchfahrenen Schlaglöcher nur eine begrenzte Zeitlang bewegen, am Anfang nur kurz, mit etwas Übung länger. Gesteuert wird das Spiel mit dem Joystick in Port 2. ■

Horoskop

Blick in die Zukunft

„Horoskop" erstellt Ihnen nach der Eingabe Ihres Geburtsdatums und eines ausgewählten Tages eine Tagesprognose und liefert sieben Glückszahlen, die Sie etwa zum Lottospielen verwenden können.

Ist etwas dran an der Astrologie? Beruhen die Voraussagen des Programmes auf astrologischen Erkenntnissen? Das Programm hat, obwohl es den Anschein erweckt, herzlich wenig mit Astrologie im Sinn. Der Programmautor sagt selbst, das Programm wäre nicht ganz ernst zu nehmen. Denn die Auswahl der Prognose erfolgt schlicht per Zufall. Doch auch zufällige Ratschläge können von Nutzen sein.

Etwas Ernstzunehmendes wurde jedenfalls verwirklicht: ein Terminkalender für wichtige Jahresdaten. Für den 6. Dezember 1989 kam etwa die Bemerkung, daß Nikolaus sei, und ob ich bereits an das Geschenk gedacht hätte. Ob Sie das Programm zum Jux benutzen oder Trost suchen, bleibt Ihnen überlassen.

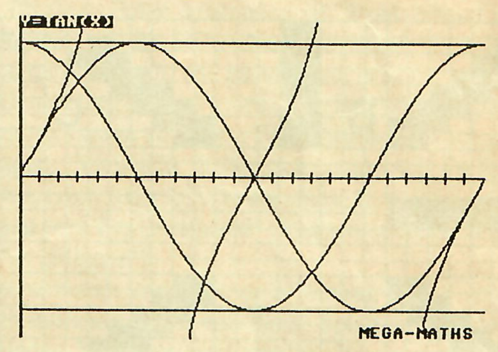
Wenn Sie nicht mehr weiterwissen und das Programm meint „Kopf hoch, es kommen auch wieder bessere Zeiten", so hat es bestimmt nicht unrecht.

Mega-Mathe

Formeln, Berechnungen, Graphen

Mega-Mathe ist mehr als eine Formelsammlung, mehr als ein Berechnungsprogramm und mehr als ein Programm zur Darstellung von Funktionen, denn es enthält alle drei Elemente zugleich und in erheblichem Umfang.

Mega-Mathe als Formelsammlung informiert über Mathematische Grundgesetze wie Assoziativ-, Kommutativ- und Distributivgesetz. Die Binomischen Formeln, das Pascalsche Dreieck, arithmetisches, harmonisches und geometrisches Mittel gehören auch zum Bestandteil der Sammlung. Eine Primzahlentabelle erweist sich als nützlich für die Bruchrechnung. Ableitungsregeln sind nötig für die Differentialrechnung, für lineare Gleichungssysteme findet sich eine Lösung. Die Geometrie, sowohl der Ebene als auch des Raumes, erfährt die ihr zukommende Beachtung.



Mega-Mathe ist auch Berechnungsprogramm. Grundrechenarten, Potenzierung und Logarithmus können ausgeführt werden. Winkelfunktionen gehören ebenfalls zum Repertoire. Für die Auswertung von Messungen erweist sich die lineare Regression mit Gleichung der Regressionsgeraden, Standardabweicheung und Korrelationskoeffizienten als nützlich. Das Berechnen von zwei- und dreidimensionalen Gebilden kann ohne Nachschlagen von Formeln bewerkstelligt werden. Weitere Funktionen sind Geraden, Ableitungen von Potentialfunktionen, Binome und Quadratische Gleichungen. Außerdem sind auch der Primzahlentest, die Prozent- und die Zinsrechnung nicht zu verachten. Zur grafischen Funktionsdarstellung wird ein weiteres Programm nachgeladen, das auch gesondert aufgerufen werden kann. Der Definitionsbereich läßt sich nicht nur als normaler Zahlenwert, sondern gar als Vielfaches von Pi festlegen. Diese Übersicht dürfte zeigen, daß Mega-Mathe für jeden, der mit der Mathematik konfrontiert wird, sich als hilfreich zu erweisen vermag.

Action-Spiele

So programmiere ich in BASIC

Bewegliche Objekte, hohe Geschwindigkeiten, großer grafischer Aufwand und gute Sound-Untermalung zeichnen Action-Spiele aus. Dem BASIC-Programmierer sind da rasch Grenzen gesetzt. Dennoch läßt sich Akzeptables erzielen, wenn fehlende Effekte durch gute Ideen ersetzt werden.

Action-Spiele sind unter den Home-Computer-Programmen wohl am verbreitetsten und beliebtesten. Sie sind dadurch gekennzeichnet, daß der Computer den Spieler zwingt, möglichst schnell auf Bildschirm-Aktionen zu reagieren. Bekannte Computerspiele wie Asteroids, Pacman, Frogger oder Defender zählen zu dieser Gattung. Im erweiterten Sinne kann man auch Simulatoren (zum Beispiel Flugsimulatoren oder Renn-Spiele) hinzunehmen.
Schon wenn Sie sich diese Beispiele vor Augen halten wird klar, daß BASIC eine sehr ungeeignete Sprache für derartige Belange ist. Dies sollte man als Handikap erkennen, es ist aber kein Grund zur Resignation. Es bedeutet lediglich, daß ich mich als BASIC-Programmierer bei Action-Spielen in einem sehr begrenzten Rahmen bewege und gewisse Abstriche hinnehmen muß:

- Action-Spiele laufen oft mit einem horizontalen oder vertikalen Scrolling ab, das heißt, die Spielfläche wandert ruckfrei in verschiedene Richtungen, wie mit einer Kamera gefilmt.

In BASIC ist nur vertikales Scrolling auf eine sehr langsame und umständliche Art möglich, so daß man meist auf derartige Vorhaben verzichten muß.

- Action-Spiele leben nicht zuletzt von einer guten Sound-Untermalung, wobei Melodien oder Geräusche eine Soundkulisse liefern, die parallel zum Spielgeschehen abläuft und so für das richtige Feeling sorgt.

Auch dieses Vorhaben ist in BASIC nur über große Umwege realisierbar und durch die Technik der Interruptsteuerung nur fortgeschrittenen Programmierern zu empfehlen.

- Action-Spiele benutzen oft bewegliche und möglichst große Figuren, die als Sprites bezeichnet werden und in Handlungsabläufe eingebunden sind. Ein Plus für das BASIC des C128 ist, daß es Sprite-Befehle kennt. Mit diesen Befehlen sollten Sie sich näher befassen, denn ansonsten ergeben sich folgende Schwierigkeiten:
  Große Figuren auf den Bildschirm zu bringen, kostet in BASIC viel Zeit.

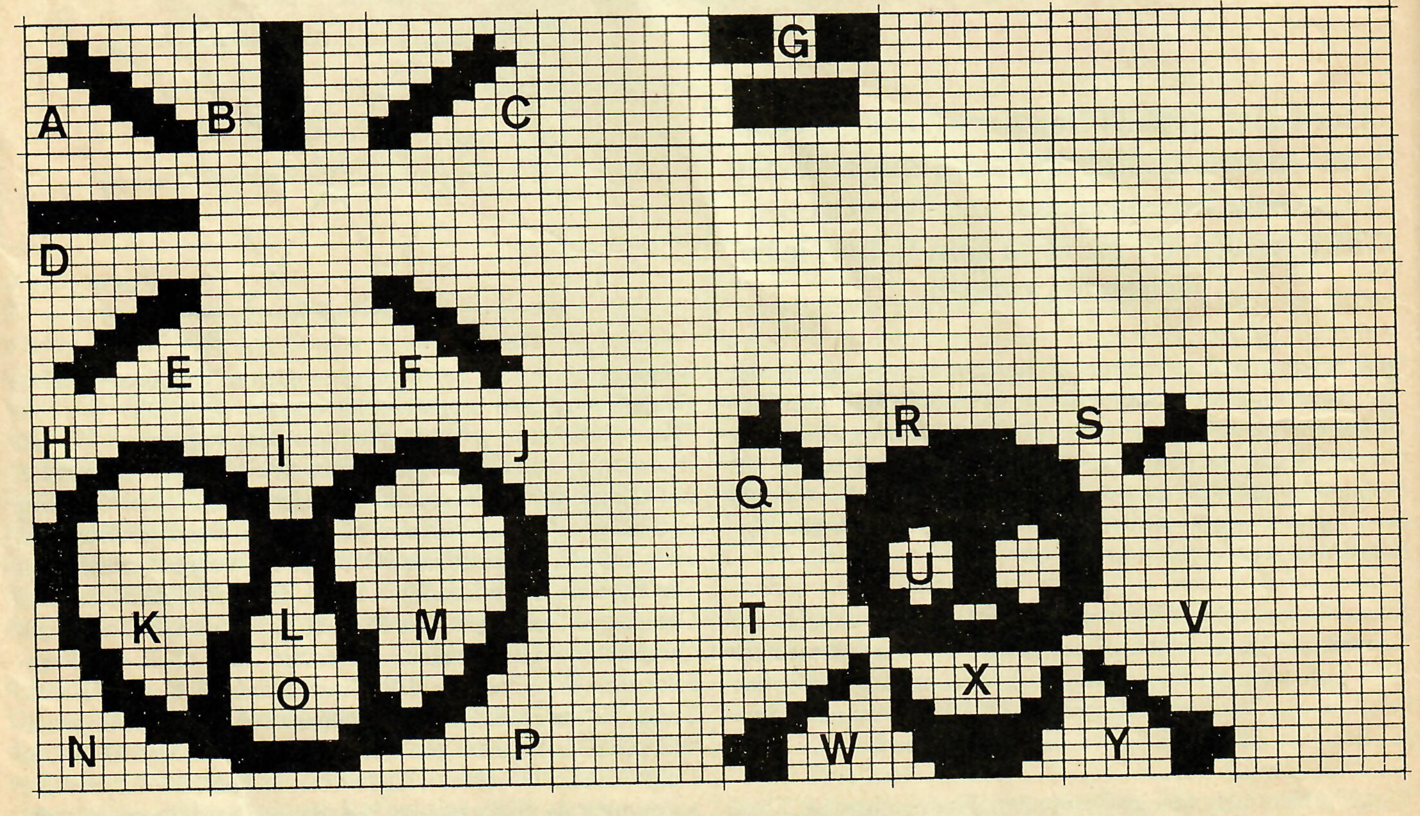


Abb.1 Diese Userzeichen finden in dem Programm Paddles Verwendung. Eigene Zeichen werden durch Kopieren des Zeichensatzes in den Arbeitsspeicher, Umändern und Umstellen der VIC-Register ermöglicht.

- Bewegungen sind in BASIC fast nur über Zeilen- und Spaltenpositionen möglich und erfolgen daher ruckhaft.
- Konsequenzen, die sich aus den Bewegungen ergeben (etwa Kollisionen) müssen rechnerisch erfaßt und in Aktionen umgesetzt werden. Auch das kostet nur allzu oft weit mehr Zeit, als es mit einem schwungvollen Spielablauf vereinbar wäre.

Mit acht bewegten Sprite-Objekten, die durch horizontale und vertikale Dehnung auch eine gehörige Größe annehmen können, läßt sich schon eine Menge anfangen. Erleichternd kommt hinzu, daß nicht ständig die Position der Sprites neu festgelegt werden muß, denn mit dem MOVSPR-Befehl kann das Sprite auch mit konstanter Geschwindigkeit in eine bestimmte Richtung auf die Reise geschickt werden. Der COLLISION-Befehl erspart die ständige Kollisionsabfrage.
Interruptgesteuert findet der Sprung in die zur Bearbeitung vorgesehene Unterroutine in dem Anblick, in dem eine Kollision erfolgt, selbständig statt.
Freilich — fehlender Hintergrundsound fehlendes Bildschirms-crolling, lediglich acht Spriteobjekt und eine Begrenzung der Spritegröße stellen eine Einschränkung unserer Möglichkeiten dar. Sprites sind auch noch lange nicht das Nonplusultra, denn sie haben eine fest definierte Gestalt. Interessant wären wohl auch Objekte, die ihre Gestalt nach bestimmten Gesetzen ständig ändern. Ein Reh, das wie eine Pappfigur über den Bildschirm geschoben wird, wirkt ziemlich unrealistisch; ein Reh, das Beine, Kopf und Körper bewegt und sich in Sprüngen fortbewegt, gefällt dem Auge des Betrachters. Unser BASIC sieht aber solche Effekte nicht vor.
Wenn Sie trotz dieser Schwierigkeiten bei BASIC bleiben wollen, bieten sich nur zwei Wege an:

1. Alle beschriebenen Merkmale und Leistungen laufen auch bei Profi-Programmen meist über Routinen ab. Das heißt, für Standard-Funktionen wie etwa Horizontal-Scrolling werden vorgefertigte Programmteile in Maschinensprache in das Programm eingebaut (ähnlich der Unterprogramm-Technik). Man kann sich quasi eine ganze Bibliothek solcher Programmteile zusammenstellen und sie wie einen Baukasten handhaben. Es ist zwar kaum möglich, aus solchen Routinen ein fertiges Programm zu basteln, aber es erleichtert die Sache ungemein.
Sowohl Programmzeitschriften als auch Vertriebsfirmen bieten solche Routinen auch für den BASIC-Programmierer an, der sie in sein Programm einbauen und von dort aus abrufen kann. Auf diese Weise lassen sich Effekte erreichen, die nur in Assembler oder Maschinensprache möglich waren. Bevor Sie von dieser Möglichkeit Gebrauch machen, sollten Sie gut BASIC programmieren können, denn es ist ein Unterschied, eine Routine nur abzurufen, oder sie in ein sinnvolles Spielgeschehen einzubinden.
Es gibt auch die Möglichkeit, BASIC-Programme in Maschinensprache umzusetzen (zu compilieren). Sie sollten sich die Anschaffung eines Compilers dennoch gut überlegen, denn er kann lediglich die Geschwindigkeit des Programms erhöhen; gewünschte Effekte wie Scrolling sind dadurch nicht erzielbar. Vielfach können die BASIC-Befehle auch nur eingeschränkt compiliert werden.

2. Beißen Sie in den sauren Apfel und ersetzen Sie fehlende Effekte durch gute Ideen!
Interessante Action-Spiele sind auch ohne technische Raffinessen möglich. Viele Programme aus der Anfangszeit des Homecomputers oder des PCs, die Sprites nicht kannten, oft gar auf jegliche Grafik verzichten mußten, da nur ASCII-Zeichen zur Verfügung standen, beweisen dies.
In der Anfangszeit der Homecomputer (1981 bis 83) gab es durchaus BASIC-Action-Programme, die zum Verkauf angeboten wurden. Es waren Spiele mit Titeln wie Snake, Bomber, 3-D-Monster-Labyrinth oder Moon-Landing, die ihr BASIC nicht verleugneten und trotzdem Spaß machten. Mein erstes gekauftes Programm (1982) hieß Nightflite und war ein BASIC-Flugsimulator. Er konnte mir mehr Wissen über Flugtechnik (zum Beispiel Radar, Leitstrahl oder Funkfeuer) vermitteln als die meisten seiner Nachfolger.
Vergessen Sie also das (eventuell) geplante BASIC-Ballerspiel oder die BASIC-Pacman-Version; so etwas können die Profis wirklich besser, und es gibt ohnehin schon genug Programme dieser Art.
Machen Sie bei Ihrer Ideensuche lieber ein paar Anleihen bei den Strategie-Spielen und setzen Sie den Spieler durch die Lösung einer schwierigen Aufgabe unter Stress! Zeitdruck, Kombinationsflut, logische Puzzles oder unübersichtliche Situationen sind hierfür bewährte Zutaten. Um die nötige Spielgeschwindigkeit zu bekommen und trotzdem stark bildschirmorientierte Handlungsabläufe vorzuweisen, ist allerdings ein Wissen nötig, das Sie im Handbuch vergeblich suchen werden. Daher soll vor allem der Umgang mit dem Bildschirm- und Farbspeicher in dieser Folge Berücksichtigung finden. Damit Ihre Spiele grafisch attraktiver werden, wollen wir Ihnen auch erklären, wie Sie eigene Grafikzeichen entwerfen und programmieren.
Wieder möchte ich Ihnen aufzeigen, wie das Beispielprogramm „Paddles" für diese Folge entstanden ist: Wohlvertraut mit den Schwierigkeiten, ein BASIC-Action-Programm zu schreiben, grübelte ich darüber nach, wie es wohl am besten zu verpacken sei, einen kleinen Gegenstand (etwa einen Ball) zielgerichtet über den Bildschirm sausen zu lassen. Etwas Strategie könnte sicher nicht schaden. Vielleicht ließe sich eine Art Fußballspiel verwirklichen (zum Beispiel in Form eines „Krökelautomaten"?) oder ein Spiel vom Typ „Arkanoid" (Ball schießt Klötzchen weg) oder eine Art „Flipper"?
Herausgekommen ist letztlich eine Mischung aus all dem: Eine automatisch laufende Kugel soll durch Spielfeldveränderungen zu wechselnden Zielen gelenkt werden. Hierzu wollte ich das Spielfeld mit variablen Umlenkflächen bestücken (sogenannte Paddles).

Als Ziele hatte ich mir Brezeln aus selbstkonstruierten Zeichen vorgestellt. Zusätzlich sollten Mauern und Totenköpfe für Erschwernisse sorgen und es ermöglichen, verschiedene Spielfelder zu gestalten.
Soweit das Prinzip; Einzelheiten können Sie der Anleitung entnehmen.

Spielanleitung für „Paddles"

Versuchen Sie eine automatisch laufende Kugel so zu beeinflussen, daß sie so viele Brezeln wie angezeigt (BR) berührt. Hierzu können Sie mit dem Joystick (Port 1) 4 mal 6 Paddles ansteuern, was an dem blauen Quadrat erkennbar ist. Ein Druck

auf die Feuertaste verändert die Stellung der Paddles. Trifft die Kugel auf einen Paddle, verändert sie ihren Lauf gemäß seiner Stellung. Der betreffende Paddle-Teil wird danach gelöscht und erscheint erst nach erneutem Stellungswechsel wieder.
Wird ein Mauerteil (rot) von der Kugel berührt, prallt sie davon ab und hinterläßt eine grüne Markierung.
Bei jedem Kontakt mit einem grünen Spielfeldanteil (Mauerveränderungen oder Totenköpfe) verlieren Sie einen Ball. Drei Bälle stehen zur Verfügung.
Wurden genügend Brezeln eingesammelt, beginnt eine neue Runde. In jeder Runde kann die Spielfläche und/oder die Spielgeschwindigkeit verändert sein. Das Spiel endet nach Verlust aller Bälle oder wenn alle Runden durchlaufen wurden.
Zur Realisierung dieser Idee war es vor allem nötig, der Kugel programmiertechnisch beizubringen, Hindernisse zu erkennen und sie farblich zu unterscheiden, um daraus die nötigen Konsequenzen zu ziehen. Für diese Vorgänge bietet es sich an, den sogenannten Bildschirm- und Farbspeicher als eine Art inneres Auge zu nutzen.
Als Bildschirmspeicher ist im RAM der Adressenbereich von 1024 bis 2047 vorgesehen. Durch in „weiß" der Computer, welches Zeichen sich auf welcher Position (Spalte/Zeile) befindet. Der Farbspeicher belegt im RAM die Adressen von 55296 bis 56319. Er speichert die Farbe jeder Bildschirmposition.
Um zu wissen, welche Adresse (POKE-Zahl) welcher Spalte/Zeilen-Kombination entspricht, zählen Sie der Anfangsadresse einer Zeile die Spaltennummer hinzu. Aus dieser Anordnung wird ersichtlich, daß horizontale Bewegungen im Farb- oder Bildschirmspeicher in Einerschritten erfolgen, verti-kale Bewegungen dagegen in Vierziger-Schritten.
Fragen Sie durch PEEK den Bildschirmspeicher ab, erhalten Sie den Bildschirm-Code. Umgekehrt bewirkt ein POKE mit einer Bildschirm-Codezahl, daß dieses Zeichen an entsprechender Stelle dargestellt wird.
Der Farbspeicher arbeitet parallel zum Bildschirmspeicher. Addieren Sie zu einem Bildschirm-POKE die Zahl 54272, befinden Sie sich im Farbspeicher an gleicher Position. Der umgekehrte Vorgang führt zum Bildschirmspeicher. Wenn Sie durch PEEK den Farbspeicher abfragen, erhalten Sie eine Zahl zwischen 0 und 255.
Die Farbinformation wird hierbei durch die vier unteren Bits repräsentiert. Die vier oberen Bits können sich ständig durch die Tätigkeit des VIC-Chips verändern und sind ohne Belang. Um den Farbwert zu erhalten, muß der durch PEEK abgefragte Wert noch mit fünfzehn undiert werden. Dadurch lassen sich die störenden oberen vier Bits beseitigen. Allerdings besteht ein kleiner Unterschied zwischen diesen Farbwerten aus dem Farbspeicher und den Farbnummern des COLOR-Befehls. Die COLOR-Farbnummern sind um den Wert eines größer als die Farbwerte des Farbspeichers.

Zur Übung:
POKE 1192,36 bewirkt, daß in Zeile 4/Spalte 8 ein Dollar-Zeichen dargestellt wird (= CHAR 1,4,8,"$")
POKE 55464,2 läßt das gleiche Zeichen in Rot erscheinen 55464 = 1192 + 54272
Versuchen Sie, in Zeile 10/Spalte 24 ein A in Grün auf den Bildschirm zu bringen.
Wenn Sie noch nie mit diesen beiden Bereichen gearbeitet haben, sind Anfangsfehler unvermeidlich.

Beispiel:
POKE 1067,32 : POKE 55339,2 ;
nichts passiert.
Durch diese beiden POKEs haben Sie ein Space in Rot in Zeile 1/Spalte 3 gesetzt. Der Farbspeicher verwaltet nur die Vordergrundfarben. Weil in dieser Hinsicht ein Space nicht viel zu bieten hat, ist es kein Wunder, daß die Auswirkungen Ihrer Befehle unsichtbar bleiben.
Poken Sie dagegen POKE 1067,13 , erscheint tatsächlich ein M in Rot.
Mit diesen Grundlagen läßt es sich schon leichter verstehen, wie die Kollision zweier Zeichen zu erfragen ist oder wie ein Ball auf dem Bildschirm verschiedene Farben auseinanderhält.
Unser Beispielprogramm Paddles macht von diesen Möglichkeiten reichlich Gebrauch. Wieder lohnt es sich, das Programm parallel zur Beschreibung, unter Einbezug der Variablenliste und der Skizze abzutippen.

Variablenliste

A,B,C	=	Universelle Variablen (z.B. für FOR-NEXT-Schleifen)
BA	=	Ballanzahl
BR	=	Brezelanzahl
BP	=	Bildschirmspeicher-POKE für Ball
DF	=	Farbspeicher—Bildspeicher
F%()	=	Spielfeld
H	=	Horizontale Feldposition
HB	=	Horizontale Brezelposition
HG	=	Horizontale Grafikposition
HS	=	Highscore
P	=	Punktanzahl
RI	=	Richtung RI=1 = Oben RI=2 = Links RI=3 = Unten RI=4 = Rechts
RM	=	Rundenmaximum
RU	=	Aktuelle Rundenzahl
SP	=	Geschwindigkeit
V	=	Vertikale Feldposition
VB	=	Vertikale Brezelposition
VG	=	Vertikale Grafikposition
X	=	Zufallszahl

Strings

A1$		
A2$		
A3$	=	Strings zum Einlesen von S$()
M$	=	Seitlicher Maueranteil
Q1$	=	Rvs ON
Q2$	=	Rvs OFF
Q3$	=	Rückpositionierung für S$()
S$()	=	Grafikzeichen S$ (1) bis S$ (4) = Paddle-Stellungen S$ (5) = Mauer S$ (6) = Brezel S$ (7) = Totenkopf S$ (8) = Space-String zum Löschen der Grafikzeichen

Mit GOTO 10000 werden im entsprechenden Unterprogramm User-Zeichen definiert (dazu später mehr).
GOSUB 9600 in Zeile 500 führt zur Dimensionierung von F%() und S$(). In Zeile 9650 werden die Steuerzeichen Rvs ON und Rvs OFF auf Q1$ und Q2$ gelegt. Q3$ verkettet mehrere Steuerzeichen, um später eine schnelle Darstellung von S$() zu erlauben.

Programmbeschreibung

GOSUB 9500 bewirkt das Einlesen von S$() *(s. Variablenliste)*. Wie die zugehörigen DATAs einzugeben sind, ist den darüberstehenden REM-Zeilen zu entnehmen.
Der Bildschirmaufbau wird durch GOSUB 9000 in Zeile 520 abgerufen:

Der seitliche Maueranteil rechts wird durch MS in Zeile 9000 dargestellt.
Innerhalb der Zeilen 9020 bis 9040 werden die Paddles gesetzt. Die Stellungen werden zufällig ermittelt.
IF H=6 THEN X = 1 in Zeile 9020 bewirkt, daß die Paddles rechts außen senkrecht stehen, was für den Balleinlauf wichtig ist.
Zeile 9050 POKEt ein Mauerstück (A,101) in roter Farbe (A+DF,2) in den Bildschirm- und Farbspeicher, um es in einer Kette am oberen und unteren Spielfeldrand erscheinen zu lassen.

Die Zeile 9060 bewerkstelligt ähnliches am linken Spielfeldrand. (In Zeile 9050 wurde der Farb-POKE aus dem Bildschirm-POKE errechnet, in Zeile 9060 ist es umgekehrt.)

Die Anzeigen werden durch Zeile 9070 und 9080 dargestellt.
GOSUB 9100 in Zeile 520 sorgt für die Spielfeldgestaltung: Ab Zeile 9100 wird abhängig von der aktuellen Rundenzahl (RU) ein RESTORE für die DATAs der jeweiligen Runde gesetzt.
Das Einlesen der DATAs ab Zeile 9200 geschieht nach folgendem Muster:
Anzahl der Zeichen für diese Runde (A).
Anzahl der Brezeln, die gesammelt werden sollen (BR).
Geschwindigkeit für diese Runde (SP).
Nummer des vorgesehenen Zeichens (bezogen auf S$()) (C)
Hierfür horizontale (HG) und vertikale (VG) Position.
Als Zeichen kommen praktisch nur Mauern (5) oder Totenköpfe (7) in Frage.
Die Zeilen 9250 bis 9260 bewirken die Darstellung der eingelesenen Zeichen in der dazugehörigen Farbe (9250 und 9255) und der zugehörigen Position (9260).

Die Zeile 530 schafft Ausgangswerte und stellt durch GOSUB 2200 den Cursor dar. Die GOSUBs 4000 bis 4400 in Zeile 540 rufen die Werte für die einzelnen Anzeigen ab. Mit GOSUB 3600 wird eine Brezel gesetzt. Dadurch, daß sich die Brezeln sowohl zwischen zwei waagerechten als auch zwischen zwei senkrechten Paddles befinden können, müssen unterschiedliche Zahlenwerte ermittelt werden (= Zeile 3610 oder 3620). Die Darstellung der Brezel erfolgt in Zeile 3630.

Die Zeilen 2000 bis 2070 bilden mit der Variablen SP eine DO-LOOP-Schleife, die erst verlassen wird, wenn alle Brezeln dieser Runde eingesammelt werden (RW=1). Die Zeilen 2010 bis 2040 sind vom Aufbau her gleich:

- Joystickabfrage unter der Bedingung, den Rand nicht zu überschreiten.
- Cursor an alter Position löschen (GOSUB 2100).
- Vertikalen oder horizontalen Wert aktualisieren.
- Cursor an neuer Position darstellen (GOSUB 2200).

Die Zeile 2050 bewirkt das Drehen der Paddles dadurch, daß jedes Paddle auf dem rechnerischen Spielfeld (F%()) seiner Stellung nach einen Wert hat, der um 1 erhöht wird. GOSUB 2300 schließt das „Überlaufen" dieses Wertes (größer als 4) aus und stellt das Paddle in seiner neuen Stellung dar.
GOSUB 300 in Zeile 2070 ermöglicht die Ballbewegung:

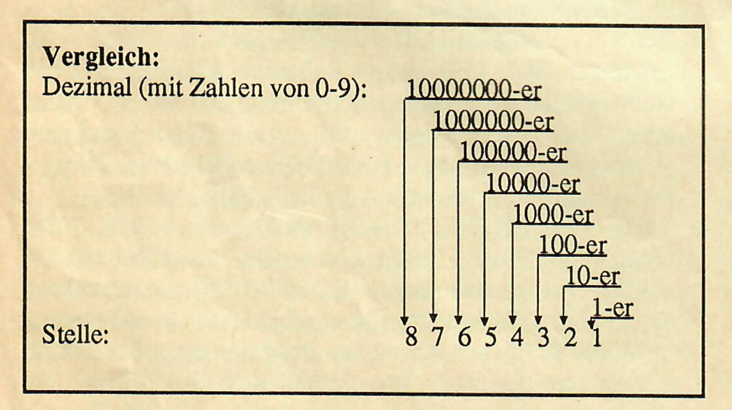
- Gemäß der aktuellen Laufrichtung RI (siehe Variablenliste) erfolgt eine Verzweigung zu den Unterprogrammen 3100 bis 3400.
  Anhand des Unterprogramms 3100 soll der grundsätzliche Aufbau erklärt werden:
- In Zeile 3100 wird durch POKE BP,32 (Space) die Kugel an alter Position gelöscht.
- Der Wert BP wird um 40 reduziert und rutscht damit im Bildschirmspeicher um eine Position nach oben.
- Wenn auf dem neu errechneten Platz ein Space zu finden ist, sorgt GOSUB 3500 für die Darstellung der Kugel an neuer Position.

Andernfalls wird in den Zeilen 3110 bis 3130 nach dem Farbwert des fremden Zeichens gefragt:

- Bei Rot (IF(PEEK(BP+DF)AND15)=2 in Zeile 3110) wird durch GOSUB 3510 das Mauerstück durch ein grünes Kreuz ersetzt; der abgezogene Wert (40) wird wieder zugezählt und die gegenteilige Richtung wird markiert (RI=3).
- Bei Grün (IF(PEEK(PB+DF)AND15)=5 in Zeile 3120) bewirkt GOSUB 3530, daß die Ballanzahl reduziert wird, daß GOSUB 4200 diese Veränderung anzeigt und daß bei Verlust aller Bälle die Rundenzahl dem Rundenmaximum gleichgesetzt wird, was in Zeile 2080 zur Endroutine führt. Das weitere Vorgehen ist mit der Zeile 3510 idenitsch.
- Bei Orange (IF(PEEK(PB+DF)AND15)=8 in Zeile 3130) ermöglicht GOSUB 3520 das Löschen der Brezel (Zeile 3520), das Setzen einer neuen Brezel (GOSUB 3600), das Anrechnen eines Punktes (Zeile 3525), die Anzeige der Punktveränderung (GOSUB 4000), den Abzug von der geforderten Brezelzahl, die Anzeige der neuen Brezelzahl (GOSUB 4100) und die Wechselmarkierung für eine neue Runde, falls alle Brezeln eingesammelt wurden.

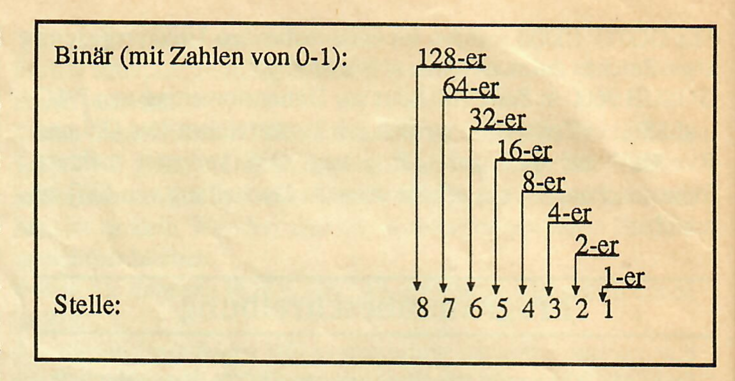
Falls keine dieser Möglichkeiten zutraf, bleiben nur die Zeichen der Paddles übrig. Als Konsequenz hierfür braucht innerhalb der Zeilen 3140 bis 3160 nur die Richtung (RI) geändert zu werden.
Wie immer soll Ihnen das Beispielprogramm viel Gelegenheit für eigene Änderungen, Verbesserungen oder Ergänzungen bieten.
Als erstes wäre vorzuschlagen, eigene Levels zu entwerfen. Hierzu müssen Sie ab Zeile 9100 ein rundenabhängiges Restore setzen,

ab Zeile 9300 die betreffenden DATAs eintragen und das Rundenmaximum (RM) in Zeile 510 aktualisieren.
Wenn es Sie stört, daß Paddles auf jeglichen Sound verzichtet, können Sie diesen Zustand ändern.
Alles sieht viel schöner aus, wenn Ihr Action-Spiel ein schönes Titelbild und eine programmierte Anleitung hat.
Vielleicht ist es möglich, das Spiel durch zusätzliche Anreize noch spannender zu machen.
Sie könnten beispielsweise die Spielregeln ändern oder zusätzliche Symbole (zum Beispiel für Extraball) ins Spiel einbauen. Nehmen Sie die Zeile 3110 oder 3130 ruhig als Mustre für eigene Abfragen. Falls Sie zu diesem Zweck eigene Zeichen entwerfen möchten, hierzu eine Anleitung.
Um eigene Zeichen zu programmieren (sogenannte User-Zeichen), ist es nahezu unerläßlich, ein wenig mehr von der Arbeitsweise Ihres C128 zu verstehen, als es bisher nötig war. Aber keine Angst, allzu schlimm wird's nicht!
Der C128 gehört wie die meisten Homecomputer zur Familie der Acht-Bit-Rechner. Ein Bit ist mit einem Lichtschalter gut vergleichbar: Entweder an oder aus = entweder eins oder null.
Die Bezeichnung Acht-Bit-Rechner rührt daher, daß der Prozessor Ihres Computers (quasi das Herzstück) gewohnt ist, mit Acht-Bit-Einheiten zu arbeiten. Diese Einheiten werden auch Bytes genannt. Normalerweise ist man es gewöhnt, sich zahlenmäßig im Dezimalsystem auszudrücken, da heißt mit einem Zahlenumfang von 0 bis 9. Der Computer muß sich dagegen mit einem Zahlenumfang von 0 bis 1 zufrieden geben. Die Rechenart, die nach diesem Prinzip arbeitet, wird als binär bezeichnet. Zu unserem Zweck ist es wichtig, binäre Zahlen in dezimale Zahlen übersetzen zu können und umgekehrt.
Das Dezimalsystem ist bekanntlich so aufgebaut, daß rechts die Einer stehen und links davon jeweils das Zehnfache der vorhergehenden Stelle (Zehner, Hunderter, Tausender und so weiter). Dieses Überlaufprinzip wird auch bei der binären Rechenart beibehalten; nur findet dieser Überlauf nicht beim Zehnfachen, sondern beim Doppelten statt.

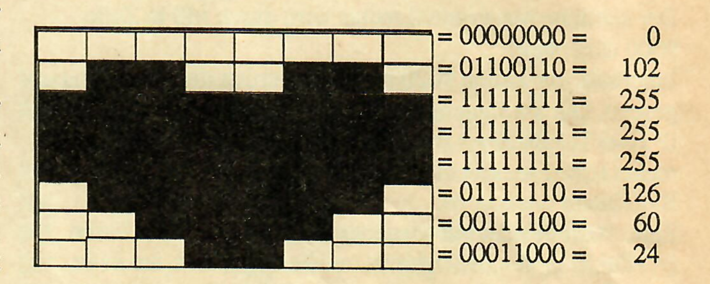


Um eine binäre Zahl ins Dezimale zu übersetzen, ist es nur nötig, den Wert einer binären Stelle zu addieren, wenn dort eine Eins angetroffen wird (von rechts nach links).

Beispiel:
10010 = 2+16 = 18
oder
11011001 = 1+8+16+64+128 = 217
Falls Sie viele binäre Zahlen dezimal übersetzen müssen, sollten



Sie nicht zögern, Ihren Computer als Arbeitspferd einzuspannen. Folgendes Mini-Programm nimmt Ihnen die Arbeit ab. Ich nehme an, daß die Funktionsweise auch ohne weitere Erklärungen einleuchtet:




```
10 INPUT "Binaer-Zahl"; BI
20 BI$ = STR$(BI) : DE = 0 : B = 1 : FOR A
= LEN(BI$) TO 2 STEP - 1
30 IF MID$(BI$,A,!)<> "1" AND MID$(BI$,A,1)
<> "0" THEN PRINT "Eingabe-Fehler!" : GOTO
10
40 IF MID$(BI$,A,!) = "1" THEN DE = DE + B
50 B = B * 2 : NEXT
60 PRINT "=" ; DE ; "Dezimal."
70 END
```

Um eine dezimale Zahl in eine binäre zu verwandeln, ist es am praktischsten, die obige binäre Stellentabelle zur Hilfe zu nehmen. Weil ein Byte acht Stellen hat, reicht die Tabelle für unsere Zwecke aus.
Wurde vorhin der Wert jeder Position addiert, ist es zur Rückübersetzung nötig, festzustellen, welche binären Wert ein der dezimalen Zahl vorhanden sind. Hierzu gehen Sie von links nach rechts vor und ziehen von der dezimalen Zahl der Reihe nach die Werte der binären Stelle ab. Falls dies möglich ist, notieren Sie an der betreffenden Stelle eine Eins, falls nicht, eine Null.

Beispiel:

233	-	128	=	105	(1.......)
105	-	64	=	41	(11......)
41	-	32	=	9	(111.....)
9	-	16	=	nicht möglich	(1110....)
9	-	8	=	1	(11101...)
1	-	4	=	nicht möglich	(111010..)
1	-	2	=	nicht möglich	(1110100.)
1	-	1	=	0	(11101001)

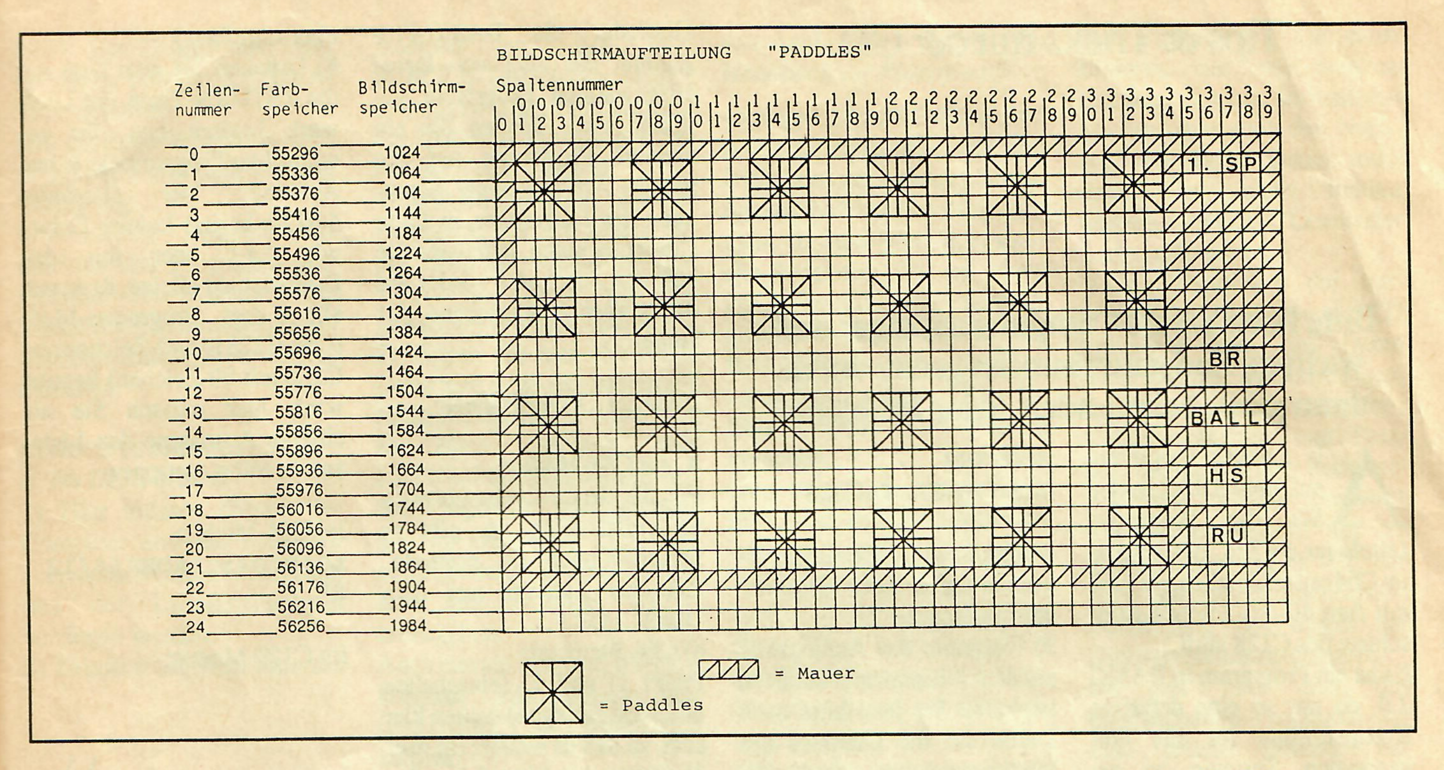


Abb.2 Bildschirmskizze für die Position der Paddles: Die dazugehörigen Adressen im Farb- und Video-RAM errechnen sich durch Addition der Spaltennummern zur jeweiligen Zeilenanfangsadresse.

Auch hierzu läßt sich ein Hilfsprogramm schreiben.
Der Zweck dieser Übungen wird schnell ersichtlich, wenn man weiß, daß jedes Zeichen des C128 aus acht mal acht Punkten besteht. Sie brauchen daher zum Entwurf eigener Zeichen ein Raster aus acht mal acht Kästchen.
Ein kleiner Haken besteht darin, daß der C128 vertikale Strukturen breiter und somit kräftiger erscheinen läßt als horizontale.
Dadurch, daß es sich bei jedem 'Zeichen um acht Zeilen mit acht Spalten handelt, kann das Raster leicht auf acht Binär-Zahlen übertragen werden.
Nach diesem System sind alle Zeichen des C128 im ROM (also im unveränderbaren Speicherbereich gespeichert.
Das Unterprogramm 10000 unseres Beispielprogramms kopiert diese Zeichen vom ROM ins RAM, um sie dort verändern zu können. Durch die GRAPHIC-Befehle in Zeile 400 wurde Platz im Bit-Map-RAM geschaffen. Zum Einlesen der von Ihnen konstruierten Zeichen dient das Ladeprogramm innerhalb der Zeilen 10370 bis 10390. Es handelt sich um eine Endlosschleife, die erst nach dem Einlesen der Zahl -1 verlassen wird. Zuerst wird jeweils der CHR$-Code des Zeichens eingelesen, das Sie verändern wollen, danach die dezimalen Übersetzungen Ihrer binären Zeichen-Konstruktion (= acht Zahlen pro Zeichen). Für diese Änderungen eignen sich am besten die Grafikzeichen (= Buchstabe plus Commodore- oder Shift-Taste); Sie können aber auch alle kleinen oder großen Buchstaben ändern.

Farbbezeichnung	Farbnr.	
Schwarz	1	0
Weiß	2	1
Rot	3	2
Cyan	4	3
Purpur	5	4
Grün	6	5
Blau	7	6
Gelb	8	7
Orange	9	8
Braun	10	9
Hellrot	11	10
Dunkelgrau	12	11
Mittelgrau	13	12
Hellgrün	14	13
Hellblau	15	14
Hellgrau	16	15

Abb.3 Color- Nummern und Poke-Codes für Farben

Alle User-Zeichen unseres Beispielprogramms sind Veränderungen der Grafikzeichen A bis Y, die in Verbindung mit der Commodore-Taste erzeugt werden.
Nach wenigen Versuchen werden Sie merken, daß diese Vorgänge nicht allzu schwierig sind. Sollten Sie nicht nur Grafik-, sondern auch Buchstaben- und Zahlzeichen verändern, so kann das Problem auftreten, daß Sie zwar im Action-Spiel auf die Originalzeichen verzichten können, nicht aber bereits im Augen- blick des Programmierens. Wenn Sie eine Berechnung durchführen möchten und es erscheinen statt Zahlen Fahrzeugteile oder ähnliches, so dürfte da unangenehm sein. Dieses Manko wird im Unterprogramm Zeile 10000 durch die Neubelegung von KEY 7 und KEY 8 ausgeglichen. KEY 8 stellt alle Zeichen in der Originalform dar. KEY 7 schaltet auf den neuen Zeichensatz um.
Wenn Ihnen die vorgestellte Methode zu umständlich ist, können sogenannte Zeichensatz-Generatoren Ihnen die Arbeit sehr erleichtern. Programme dieser Art können sowohl gekauft als auch aus Computerzeitschriften abgetippt werden. Sie ermöglichen das automatische Übersetzen der Zeichenkonstruktion in die entsprechenden DATAs und haben meist einige Extras wie Spiegelung, Drehung oder inverse Darstellung der Zeichen zu bieten.
In jedem Fall dürfte nun einer grafischen Auffrischungskur Ihrer Programme nichts mehr im Wege stehen.

Peter Bergenlam

Tips & Tricks mit der CD

Kein Problem

Bitte lesen Sie als Computeranfänger diese wichtige Ladeanweisungen genau durch, dann gibt's beim Laden keine Probleme!

Als zweites Programm auf Ihrer DISC finden Sie das Programm *Disclader 64/128*, das sowohl im C64 mit 40 Zeichen, als auch mit dem 40-/80-Zeichen-Bildschirm des C128 läuft.
Es ist ein Programm in BASIC 2.0, da nur so eine derartige Kompatibilität für alle drei genannten Systeme zu erreichen war.
Folgende Punkte sollten aber noch immer beachtet werden:
1) Schalten Sie Computer und Floppy ein.
2) Legen Sie die CD in den Laufwerksschacht und verriegeln Sie diesen. Jetzt richtet sich Ihr weiteres Vorgehen danach, ob Sie C64- oder C128-Programme laden möchten. Als C64-Benutzer geben Sie bitte ein:

```
LOAD":*",8,1
```

Damit lädt der C64 zuerst das Maschinenfile *Boot.64*, das sich an erster Stelle auf Ihrer Diskette befindet, das seinerseits wieder den *Disclader 64/128* nachlädt und auch sofort startet.
Besitzen Sie einen C128 und befinden sich auch in diesem Modus, genügt es normalerweise, wenn Sie kurz den RESET-Taster (seitlich rechts an Ihrem Gerät) betätigen oder den Befehl BOOT eingeben und dann die RETURN-Taste drücken.
Beides erzeugt denselben Effekt, auf Ihrem Bildschirm erscheint die Meldung:

```
BOOTING...
DISCLADER 64/128
```

Auch hier wird dieses Ladeprogramm nach dem „Booten" sofort gestartet.
3) Nachdem das Anfangsbild auf dem Bildschirm erschienen ist, rufen Sie jetzt bitte nach Druck auf die Leertaste das Inhaltsverzeichnis, das *DIRECTORY*, der gerade aktuellen CD auf.
4) Die Liste der Programme, die sich auf der DISC befinden, werden nun aufgelistet, vor den Programmnamen erscheint ein kleiner Pfeil, den Sie mit den Cursortasten auf und ab bewegen können.
Als C64-Benutzer sollten Sie die mit diesem Kennzeichen beginnenden Programmnamen auswählen, wenn Sie sich im 128er-Modus befinden, dann nur solche. Das Laden von Programmen jeweils in den nicht dafür vorgesehenen Computertyp ergibt in den meisten Fällen nur Unsinn.
Bei den C128-Programmen ist zusätzlich noch in Klammern angegeben, ob Sie für den 40- oder 80-Zeichen-Modus gedacht sind, als (40) oder (80).
5) Haben Sie sich für ein entsprechendes Programm für den zutreffenden Computermodus entschieden, positionieren Sie den Pfeil vor dessen Namen und drücken einfach die RETURN-Taste.
6) Das gewünschte Programm wird jetzt automatisch geladen und sofort gestartet.
Wie bereits erwähnt, ist dieser Disclader ein BASIC-Programm, das aber von Ihrem ausgewählten Programm gelöscht wird. Wollen Sie ein anderes Programm auf die gleiche Art, also unter Benutzung des *Discladers*, in Ihren Computer holen, so müssen Sie halt den *Disclader* wieder auf die vorher beschriebene Art laden.

LADEN „VON HAND"

A. Im C64-Modus
Für alle, denen dies immer noch zu umständlich ist, gibt es natürlich noch die „normale" Art des Ladens: aus dem DIRECTORY.
a) C64-Benutzer:
Punkt 1) und 2) (einschalten und Disc einlegen) sollten klar sein, so daß es weitergeht mit:
3) Geben Sie bitte folgenden Befehl ein:

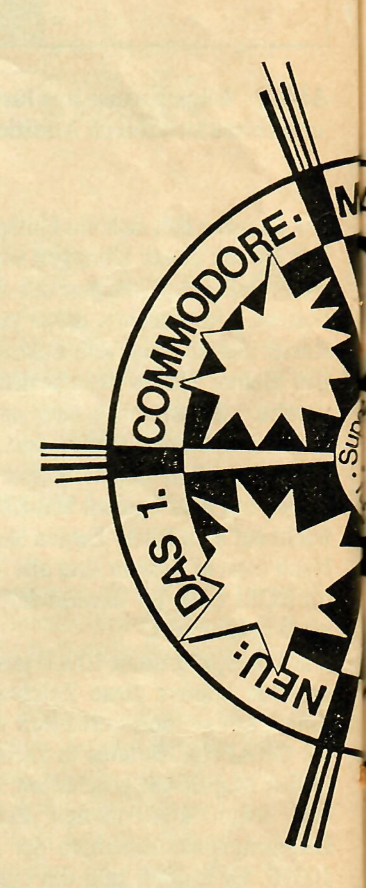
```
LOAD"$",8
```

und drücken Sie die RETURN-Taste. Durch die Bezeichnung "$" wird das neue Diskinhaltsverzeichnis der CD (oder jeder anderen Diskette) geladen.
4) Nachdem sich der Computer wieder mit READY meldet, tippen Sie bitte LIST ein und drücken wieder die RETURN-Taste. Auf dem Bildschirm wird jetzt das genaue Inhaltsverzeichnis Ihrer CD aufgelistet.
5) Entscheiden Sie sich für das Programm, das Sie laden möchten.
6) Gehen Sie mit dem Cursor nach oben, bis dieser in der Zeile vor dem gewählten Programmnamen steht (an die Stelle, an der die Zahl der belegten Blocks eingetragen ist).
7) Geben Sie an dieser Stelle den Befehl LOAD ein. Bitte achten Sie darauf, daß das erste „Gänsefüßchen" vor dem Programmnamen nicht überschrieben wird! (Die Zahl, die die vom Programm belegten Blökke angibt, dürfen Sie ruhig überschreiben.)
8) Drücken Sie jetzt bitte nur die RETURN-Taste, das Laufwerk beginnt nun, Ihr gewünschtes Programm zu laden.
Achtung: Der *Disclader 64/128* besitzt keinen Zusatz wie ',8:' oder nur den Doppelpunkt alleine, da er von allen drei Programm-Modi (C64/C128 mit 40-Zeichen und C128 mit 80-Zeichen) benutzt wird, hier müssen Sie als einzige Ausnahme den Ladezusatz selbst eintippen.

Im C64-Modus:

```
LOAD"DISCLADER64/12
8",8:
```

Im C128-Modus:




```
RUN"DISCLADER":
```

9) Wenn sich nach dem Ladevorgang, bei dem SEARCHING FOR... und LOADING vom Computer als Arbeitsmeldung auf dem Bildschirm ausgegeben werden, der blinkende Cursor mit READY wieder meldet, so tippen Sie dort ein:

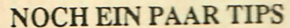
RUN: (den Doppelpunkt nicht vergessen und die RETURN-Taste drücken).
Das geladene Programm wird so gestartet und steht zu Ihrer Verfügung.
Ein kleines Beispiel: Sie möchten das Programm „Eisenbahn" von Diskette laden. Die exakte Ladeanweisung auf dem Bildschirm müßte dann so aussehen:

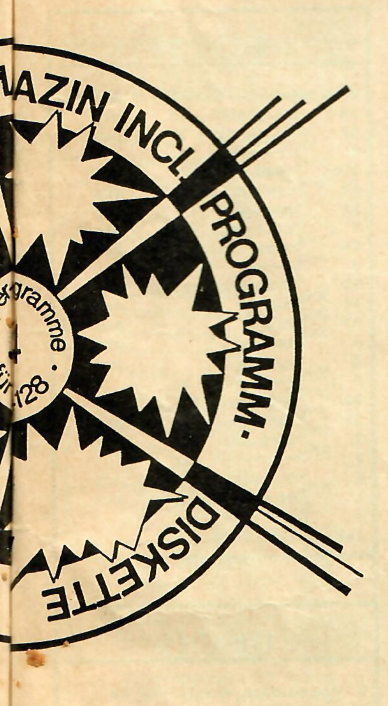
```
LOAD"EISENBAHN",8
```

B. Im C128-Modus

1) Hier müssen ebenso der Computer und das Laufwerk eingeschaltet, die CD eingelegt und der Laufwerksschacht verriegelt werden.
2) Drücken Sie jetzt bitte die Funktionstaste "F3" (oben rechts, über der Tastatur).
Diese Taste ist schon beim Einschalten des 128PC mit dem Befehl zum Aufruf des Disketteninhaltsverzeichnisses belegt (DIRECTORY) und wird auch sofort ausgeführt.
3) Stoppen können Sie die Ausgabe auf dem Bildschirm jederzeit mit der NO-SCROLL-Taste (oben Mitte), den ungewollten Ladevorgang des DIRECTORY brechen Sie mit der STOP-Taste ab.
4) Wählen Sie auch hier ein Programm aus, und fahren Sie mit Hilfe der entsprechenden Taste den Cursor vor den gewählten Programmnamen.
5) Tippen Sie hier ein: RUN (auch hier kann die Blockzahl ruhig überschrieben werden).
6) Nach Drücken von RETURN läuft die Floppy an, das Programm wird geladen und im Gegensatz zum C64 sofort gestartet, es erübrigt sich also, hier nochmals "RUN" einzugeben.
Auch hier ein Beispiel, wie eine korrekte Ladeanweisung aus dem DIRECTORY beim C128 aussieht:

```
RUN"CHECK-LISTE":
```





NOCH EIN PAAR TIPS

➩ Als C64-Benutzer können Sie (allerdings immer nur das erste File, das sich auf Ihrer Diskette befindet, und dazu muß es auch unbedingt ein Programm = PRG-File sein) mit Druck auf zwei Tasten in den Commodore 64 laden: SHIFT/RUNSTOP. Es startet dann sofort.

➩ Wenn's nicht das erste File auf der Disk ist, wird die Ladeanweisung etwas umfangreicher: Siehe soeben behandelte Ladeanweisungen!

➩ Files, die hinter dem Namen nicht die Bezeichnung "PRG" eingetragen haben, etwa SEQ, REL, DEL und USR lassen sich mit diesen besprochenen Ladeanweisungen weder laden noch starten. Es handelt sich hier um reine Datenansammlungen von Byte, die nur vom dafür gedachten Hauptprogramm (egal, ob in BASIC oder Maschinensprache) wieder in den Computerspeicher geladen und dort verarbeitet werden können. *Lesen Sie dazu bitte die entsprechenden Seiten im Handbuch zur Floppy nach.*

➩ Nicht alle PRG-Files sind BASIC-Programme, die mit RUN gestartet werden können. Sehr oft handelt es sich hier um nicht listbare Maschinensprache-Routinen, die an einen ganz anderen Speicherplatz geladen werden, als ihn ein BASIC-Programm benötigt, zum Start würden Sie die Eingabe eines "SYS"-Befehls brauchen.
Oder es handelt sich auch hier um eine reine Byte-Sammlung wie etwa der Inhalt eines Sprite-Speichers oder eines hochauflösenden Grafikbildes. Diese Files werden ebenfalls in der Regel von einem Haupt-Steuerprogramm zur Verarbeitung benötigt und bei Bedarf während des Programmablaufs nachgeladen.
Ein Blick genügt: Alle PRG-Files auf der CD, die entweder mit dem Zusatz ',8: beziehungsweise ‚8,1:' (bei den C128-Programmen nur mit Doppelpunkt) gekennzeichnet wurden, sind lade- und ausführbare PRG-Files, alle anderen nur Daten, die von einem Hauptprogramm nachgeladen werden müssen.

➩ Die Zahl vor den Filenamen im DIRECTORY ist die Anzahl der belegten Blocks. Möchten Sie (in etwa) wissen, wieviele Byte so ein Programm im Speicher Ihres Computers belegen würde, so multiplizieren Sie diese Zahl mit „256".

➩ Die Meldung am Ende des DIRECTORY "XX BLOCKS FREE" gibt Ihnen darüber Auskunft, wieviel Platz (in Blöcken gerechnet) sich noch auf der aktuellen Diskette befindet. Wollen Sie die genaue Byteanzahl wissen, so müssen Sie diese Zahl ebenfalls mit „256" malnahmen.

➩ Die CD ist (normalerweise) dadurch schreibgeschützt, daß sich auf der rechten oberen Seite neben dem Etikett keine Schreibkerbe befindet. Das hat den Vorteil, daß Sie diese Diskette nie versehentlich löschen oder neu formatieren können, aber auch den Nachteil, daß keine Daten mehr darauf gespeichert werden können. (Von DISC Lesen oder Laden geht immer.)
Viele Programme der verschiedensten Ausgaben der CD verlangen aber, daß Daten auch abgelegt werden können, denken Sie nur an Anwender-, Dateiverwaltungs- und Textprogramme. Ja, aber auch verschiedene Spiele brauchen einen Schreibzugriff auf die Diskette, und sei es nur, das neue, aktuelle High-Score nach Beendigung des Spiels auf Diskette zurückzuschreiben.
Da gibt's nur eins: das entsprechende Programm auf eine andere Diskette *mit* Schreibkerbe zu speichern beziehungsweise diese Files einzeln mit einem entsprechenden Kopierprogramm rüberzukopieren oder sich gleich mit einem Kopier-Tool für ganze Disketten ein „Backup" Ihrer CD auf eine schreibfähige Disk zu ziehen. Oder Sie benutzen einen Diskettenlocher und bringen damit die entsprechende Schreibkerbe an. Wir empfehlen auf alle Fälle die Kopiermethode.

KOPIERSCHUTZ — NEIN, DANKE!

Wir haben es bisher nicht getan und werden es auch künftig nicht tun: einen Kopierschutz auf der CD installieren. Nicht, weil wir es nicht könnten, sondern weil wir keinen Sinn darin sehen.
Und schließlich ist doch die Tatsache, daß sich Raubkopierer strafbar machen, nicht erst seit heute bekannt. Jeder soll die Möglichkeit haben, sich so viele Sicherheitskopien seiner Disketten zu eigenen, privaten Zwecken zu machen, wie er es für richtig hält.

Und nun viel Spaß mit der neuen COMMODORE DISC!

Haben Sie schon alle CDs?

COMMODORE DISC LABELS

Ordnung und Übersicht

Für Leute dich sich einen Überblick schaffen wollen, welche Programme auf welcher **COMMODORE DISC** zu finden sind, denen kann geholfen werden. Von CD Nummer 01 bis CD Nummer ff werden wir in loser Folge alle Disketten-Inhaltsverzeichnisse als Labels zum Aufkleben auf die Diskettentasche veröffentlichen. Sie dienen auch zur Information, um die in Ihrer Sammlung fehlende Ausgaben der CD ergänzen zu können. Schreiben Sie die Ihnen fehlenden bzw. gewünschten Ausgaben der CD, auf den Serviceseiten des Heftes gedruckten **COMMODORE -DISC -ERGÄNZUNGS-COUPON** und schicken ihn an uns. Die gewünschten Ausgaben erhalten Sie dann schnellstmöglich.

DISC 1

```
0    "commodore disc 1" 01 2a
0       "----------" usr
0       "commodore disc 1" usr
0       "----------" usr
138     "john bend      64" prg
64      "koenig arthus 64" prg
26      "adressen-reg. 64" prg
17      "vokabelpauker 64" prg
2       "test"            seq
0       "----------" usr
120     "john bend     128" prg
71      "koenig arthus128" prg
32      "textprogramm 128" prg
0       "----------" usr
0       "(c) copyright by" usr
0       "ca verlags gmbh " usr
0       "----------" usr
185  blocks free.
```

DISC 2

```
0    "commodore disc 2" 02 2a
0       "----------" usr
0       "commodore disc 2" usr
0       "----------" usr
58      "kalorientab.  64" prg
66      "kalo/tabelle"    seq
40      "kalo/kalorien"   seq
44      "kalo/eiweiss"    seq
51      "kalo/fett"       seq
60      "kalo/hydrate"    seq
16      "polit-phrasen 64" prg
26      "sechs verliert64" prg
26      "adressen richtig" prg
0       "----------" usr
43      "kaufmann     128" prg
47      "fotografie   128" prg
25      "sechsverliert128" prg
16      "polit-phrasen128" prg
0       "----------" usr
0       "(c) copyright by" usr
0       "ca verlags gmbh " usr
0       "----------" usr
146  blocks free.
```

DISC 3

```
0    "commodore disc 3" 03 2a
3       "einleitung     " prg
0       "----------" usr
0       "commodore disc 3" usr
0       "----------" usr
100     "yahtzee       64" prg
50      "spy (spion)   64" prg
38      "lawine        64" prg
25      "diskeditor    64" prg
48      "mathehilfe    64" prg
30      "textprogramm  64" prg
0       "----------" usr
61      "spielautomat 128" prg
32      "biorhythmus  128" prg
0       "----------" usr
0       "(c) copyright by" usr
0       "ca verlags gmbh " usr
0       "----------" usr
263  blocks free.
```

DISC 4

```
0    "commodore disc 4" 04 2a
3       "einleitung"      prg
0       "----------" usr
0       "commodore disc 4" usr
0       "----------" usr
11      "maserati......64" prg
25      "1764..........64" prg
39      "cosmiccarnage 64" prg
10      "floppy-fehler.64" prg
16      "funkt.tasten..64" prg
54      "ges.directory.64" prg
0       "----------" usr
59      "graue moewe..128" prg
27      "musicmaschine128" prg
53      "telemuehle...128" prg
34      "datei(80z)...128" prg
0       "----------" usr
0       "(c) copyright by" usr
0       "ca verlags gmbh " usr
0       "----------" usr
330  blocks free.
```

DISC 5

```
0    "commodore disc 5" 05 2a
22      "disclader.....64" prg
15      "disclader....128" prg
0       "----------" usr
38      "django........64" prg
56      "composer......64" prg
5       "comp mc"         prg
16      "silent running"  prg
16      "wishing"         prg
67      "taktik........64" prg
3       "asc1"            prg
10      "asc2"            prg
7       "asc3"            prg
2       "asc4"            prg
7       "char"            prg
1       "clear"           prg
16      "map"             prg
4       "sprites"         prg
31      "zeicheneditor.64" prg
26      "schoenschrift.64" prg
5       "multi-screen..64" prg
0       "----------" usr
30      "prodat.......128" prg
38      "paradies.....128" prg
65      "pelon........128" prg
0       "----------" usr
0       "(c) copyright by" usr
0       "ca verlags gmbh " usr
0       "----------" usr
184  blocks free.
```

DISC 6 A/B

```
0    "commodore disc 6" 06 2a
23      "disclader.....64" prg
15      "disclader....128" prg
0       "----------" usr
97      "antimonopoly..64" prg
75      "a-z quiz......64" prg
33      "comtape.......64" prg
70      "roulette......64" prg
22      "user-help.....64" prg
0       "----------" usr
45      "oil company..128" prg
156     "monopoly.....128" prg
102     "multidatei...128" prg
0       "----------" usr
0       "(c) copyright by" usr
0       "ca verlags gmbh " usr
0       "----------" usr
0       "auf der seite 2 " usr
0       "supergame disc 6" usr
0       "----------" usr
0       "commodore disc 6" usr
0       "----------" usr
26   blocks free.
0    "commodore disc 6" 06 2a
12      "boot"            prg
7       "loader"          prg
28      "menue"           prg
52      "c/game"          prg
54      "high score"      prg
55      "erklaerung"      prg
1       "page4.neu"       prg
1       "copy"            prg
9       "sprite"          prg
2       "gp"              prg
1       "bild holen"      prg
1       "bild zurueck"    prg
1       "auto"            prg
1       "h - s"           seq
16      "lade bild"       prg
16      "bild h-s"        prg
33      "bild 1"          prg
33      "bild 2"          prg
33      "bild 3"          prg
33      "bild 4"          prg
33      "bild 5"          prg
33      "bild 6"          prg
2       "lade level"      prg
2       "level 1"         prg
2       "level 2"         prg
2       "level 3"         prg
2       "level 4"         prg
2       "level 5"         prg
2       "level 6"         prg
2       "level 7"         prg
2       "level 8"         prg
2       "level 9"         prg
0       "----------" usr
```

DISC 7

```
0    "commodore disc 7" 07 2a
23      "disclader.....64" prg
15      "disclader....128" prg
0       "----------" usr
4       "loisaida......64" prg
38      "main"            prg
15      "dat"             prg
2       "caverns loader"  prg
11      "caverns part ii" prg
26      "caverns part iii" prg
4       "caverns music"   prg
9       "caverns sprites" prg
105     "caverns of death" prg
4       "caverns part iv" prg
1       "caverns mc"      prg
28      "general spr.ut.." prg
29      "gen.spr.ut..demo" prg
3       "bytes free....64" prg
3       "merge.........64" prg
0       "----------" usr
103     "mastermind...128" prg
44      "ligatabelle..128" prg
2       "merge........128" prg
1       "text 80 load.128" prg
35      "t801"            prg
123     "Text 80 compil128" prg
9       "27 zeilen    128" prg
0       "----------" usr
0       "(c) copyright by" usr
0       "ca verlags gmbh " usr
0       "----------" usr
0       "commodore disc 7" usr
0       "----------" usr
27   blocks free.
```

DISC 10

```
0    "commodore disc10" 10 2a
23      "disclader.64"    prg
15      "disclader.128"   prg
0       "----------" usr
113     "disc box.64"     prg
17      "mathematik.64"   prg
82      "moonbase.64"     prg
7       "file-copy.64"    prg
23      "crash race.64"   prg
0       "----------" usr
43      "star-trotter.128" prg
33      "pic 1"           prg
33      "pic 2"           prg
3       "sprite 1"        prg
3       "sprite 2"        prg
0       "----------" usr
84      "boersenspiel.128" prg
21      "datagen.128"     prg
2       "cent.hex"        prg
0       "----------" usr
1       "(c) copyright by" prg
1       "ca - verlag gmbh" prg
0       "----------" usr
160  blocks free.
```

DISC 11

```
0    "commodore disc11" 11 2a
19      "disclader.64"    prg
9       "disclader.128"   prg
0       "----------" usr
67      "wordman.64"      prg
1       "mob-designer"    prg
44      "mob.main"        prg
25      "videodatei"      prg
109     "verschollen i.n." prg
31      "dubble fighter"  prg
0       "----------" usr
27      "high-draw-128"   prg
3       "pfeile"          prg
89      "bilanzermittlung" prg
46      "vokabel-prg.128" prg
0       "----------" usr
1       "(c) copyright by" prg
1       "ca - verlag gmbh" prg
0       "----------" usr
191  blocks free.
```

SERVICE

DISC 8

```
0   "commodore disc 8" 08 2a
22    "disclader.64"     prg
15    "disclader.128"    prg
0     "————————" usr
12    "time step.64"     prg
37    "devisen.64"       prg
83    "nuklear.64"       prg
15    "basic v.2+.64"    prg
29    "1.demo v2+"       prg
30    "2.demo v2+.64"    prg
13    "intern.lader.64"  prg
33    "internat.prg"     prg
6     "titelbild"        prg
0     "————————" usr
83    "conquest.128"     prg
3     "spr.1.et's 1"     prg
3     "spr.2 all 2"      prg
3     "spr.3.et's 2"     prg
3     "spr.4.et's 3"     prg
3     "spr.5.et's 4"     prg
37    "conquest.title"   prg
69    "kniffel.128"      prg
102   "weltall.128"      prg
0     "————————" usr
0     "(c) copyright by" usr
0     "ca-verlags  gmbh" usr
0     "————————" usr
63  blocks free.
```

DISC 9

```
0   "commodore disc 9" 09 2a
23    "disclader.64"     prg
15    "disclader.128"    prg
0     "————————" usr
6     "basic-tool.64"    prg
42    "disk-tool.64"     prg
73    "schalter.64"      prg
0     "————————" usr
1     "supermon.64"      prg
1     "monitor.stack"    prg
28    "monitor.main"     prg
0     "————————" usr
39    "bundesliga.64"    prg
7     "titel"            prg
1     "tab"              prg
3     "1.bundesliga"     seq
3     "2.bundesliga"     seq
0     "————————" usr
1     "karteikasten"     prg
6     ".1-cardbox"       prg
5     ".d-cardbox"       prg
37    ".m-cardbox"       prg
1     ".dat-cardbox-1"   prg
1     ".dat-cb.-1 alt"   prg
1     ".dat-cardbox-2"   prg
1     ".dat-cb.-2 alt"   prg
1     ".dat-cardbox-3"   prg
1     ".dat-cb.-3 alt"   prg
1     ".dat-cardbox-4"   prg
1     ".dat-cb.-4 alt"   prg
1     "leerdatei"        prg
0     "————————" usr
56    "diskcall.128"     prg
55    "Franz-Trainer"    prg
6     "Unit  1"          seq
8     "Unit  2"          seq
6     "Unit  3"          seq
3     "Unit  4b"         seq
7     "Unit  4a"         seq
4     "Unit  4c"         seq
4     "Unit  4d"         seq
0     "————————" usr
1     "(c) copyright by" prg
1     "ca-verlag  gmbh"  prg
0     "————————" usr
213  blocks free.
```

DISC 12

```
0   "commodore disc12" 12 2a
18    "disclader.64"     prg
8     "disclader.128"    prg
0     "————————" usr
52    "quiz-master v3.8" prg
22    "qm zeichensatz"   prg
30    "dateigenerator"   prg
2     "qm loader"        prg
2     "qm music irq mc"  prg
4     "qm music mc"      prg
1     "qm sprite irq mc" prg
5     "qm sprites mc"    prg
2     "qm garbage mc"    prg
27    "wm 78-86.dat"     seq
23    "geschichte.dat"   seq
1     "wm 78-864.dat"    seq
0     "————————" usr
39    "ekst.10e"         prg
56    "aktienspiel"      prg
0     "————————" usr
4     "  Menuecalc V1.6" prg
1     "(C)1987 Ch.Braun" prg
41    "  Mps 801/803"    prg
41    "  Star NL 10 "    prg
1     "  Drucker     "   seq
3     "fruehstueck"      seq
3     "gulasch"          seq
3     "bolognese"        seq
4     "schweinelendchen" seq
41    "menuecalc"        prg
5     "  lader2"         prg
42    "  Star NL 10 +"   prg
0     "————————" usr
61    "scriptcall"       prg
1     "sc 1"             prg
3     "sc 2"             prg
9     "sc 3"             prg
4     "sc 4"             prg
5     "sc 5"             prg
10    "sc 6"             prg
1     "sc 7"             prg
5     "install epson"    prg
0     "————————" usr
17    "calc128"          prg
0     "————————" usr
1     "(c) copyright by" prg
1     "ca - verlag gmbh" prg
0     "————————" usr
65  blocks free.
```

DISC 14

```
0   "commodore disc14" 14 2a
18    "disclader.64"     prg
8     "disclader.128"    prg
0     "————————" usr
31    "scs-basic"        prg
10    "scs.demo"         prg
2     "bloodmon"         prg
12    "bmmas"            prg
64    "bmass"            prg
4     "lachmann"         prg
81    "lm.mapro.v1.0"    prg
0     "————————" usr
33    "picture-show"     prg
37    "psm. 00"          prg
37    "u.f. 00"          prg
37    "u.f. 02"          prg
37    "u.f. 04"          prg
46    "buchhaltung"      prg
1     "vdc-basic"        prg
9     "vdc-basic.main"   prg
4     "vdc-demo"         prg
1     "vdc-demo2"        prg
37    "terminkalender"   prg
1     "Start Uni-Dat II" prg
43    "Uni-Dat II Prog." prg
1     "Uni-Dat II + Esc" prg
0     "————————" usr
1     "(c) copyright by" prg
1     "ca - verlag gmbh" prg
0     "————————" usr
107  blocks free.
```

DISC 13

```
0   "commodore disc13" 13 2a
18    "disclader.64"     prg
9     "disclader.128"    prg
0     "————————" usr
79    "the red planet"   prg
114   "red planet main"  prg
31    "crazy car"        prg
39    "cc spr+masch dat" prg
6     "cc maschine"      prg
4     "cc sprites"       prg
16    "basic-editor"     prg
1     "lader"            prg
0     "————————" usr
120   "kassa-programm"   prg
3     "m.code"           prg
7     "umlaute im vdc"   prg
9     "high-low.print"   prg
5     "farbram.v1.1"     prg
2     "farbram.demo"     prg
30    "berechn.fl.&vol." prg
0     "————————" usr
1     "(c) copyright by" prg
1     "ca - verlag gmbh" prg
0     "————————" usr
168  blocks free.
```

DISC 15

```
0   "commodore disc15" 15 2a
18    "disclader.64"     prg
9     "disclader.128"    prg
0     "————————" usr
9     "agent"            prg
150   "0013"             prg
8     "breaker"          prg
23    "imperium"         prg
12    "romanum"          prg
65    "ca - graf"        prg
5     "ca - movie show"  prg
17    "ca - rout 1"      prg
9     "ca - rout 2"      prg
4     "ca menu ctrl"     prg
30    "UFO"              seq
26    "Glas"             seq
64    "Haus2.f"          prg
0     "————————" usr
10    "zeichensatz(prg)" prg
1     "zeichensatzholer" prg
1     "ram-rom"          prg
17    "zeichens."        prg
9     "anleitung"        prg
30    "textverarbeitung" prg
24    "funktionsgenie"   prg
60    "pc-grafik"        prg
0     "————————" usr
1     "(c) copyright by" prg
1     "ca-verlags  gmbh" prg
0     "————————" usr
61  blocks free.
```

DISC 19

```
0   "commodore disc19" 19 2a
1     "boot.64",8,1:     prg
9     "disclader64/128"  prg
0     "————————" usr
40    "assemb.kurs3",8:  prg
35    "mc-routinen 3"    prg
56    "diskmaster",8:    prg
3     "renumber",8,1:    prg
3     "w-text",8:        prg
115   "w-t.prg"          prg
0     "————————" usr
5     "buchungen":       prg
94    "-Buch.PRG.      " prg
4     "notizzettel":     prg
48    "-Noti.PRG.      " prg
9     "zei0            " prg
9     "zei1            " prg
81    "multi play":      prg
98    "mofa-test":       prg
0     "————————" usr
0     "(c) copyright by" usr
0     "ca - verlag gmbh" usr
0     "————————" usr
54  blocks free.
```

DISC 16

```
0   "commodore disc16" 16 2a
19    "disclader.64"     prg
9     "disclader.128"    prg
0     "————————" usr
12    "autoboot-maker"   prg
72    "schiff versenken" prg
29    "crash"            prg
15    "master-c. 3.1"    prg
2     "demo"             prg
17    "graphik-lader"    prg
40    "graphik 1"        prg
40    "graphik 2"        prg
17    "goldfever"        prg
30    "gf/m"             prg
4     "gf/t"             prg
1     "gf/score"         seq
0     "————————" usr
46    "mensch aergere"   prg
35    "benzinkosten"     prg
29    "rechnung"         prg
10    "boot-starter"     prg
84    "kapital"          prg
42    "kalender"         prg
34    "suchdatei"        prg
37    "terminkal.v 2.0"  prg
0     "————————" usr
1     "(c) copyright by" prg
1     "ca -verlags gmbh" prg
0     "————————" usr
37  blocks free.
```

DISC 23

```
0   "commodore disc23" 23 2a
1     "boot.64",8,1:     prg
11    "disclader64/128"  prg
0     "——c 64:——" usr
6     "graphbasic",8:    prg
4     "demo.malprg",8:   prg
15    "demo.graph",8:    prg
0     "————————" usr
35    "dubbledate",8:    prg
7     "text"             seq
1     "zahl/text"        seq
0     "————————" usr
1     "action-sys",8,1:  prg
14    "initial"          prg
9     "chst1"            prg
9     "chst2"            prg
48    " action-creator " prg
1     "  action-basic  " prg
17    "bc1.o"            prg
14    "bc2.o"            prg
6     "a.b.-demo",8:     prg
30    "action-character" prg
49    "  action-datas  " prg
0     "————————" usr
4     "pro menu control" prg
6     "pmc demo",8:      prg
0     "————————" usr
2     "diskmerge",8:     prg
0     " 128 pc: " usr
124   "phildat":         prg
1     "Testsammlung 1.i" seq
13    "Testsammlung 1.p" seq
0     "————————" usr
18    "solitaire":       prg
126   "geo calculate":   prg
9     "apfelgrafik":     prg
0     "————————" usr
7     "weltzeit":        prg
32    "weltkarte"        prg
0     "————————" usr
0     "(c) copyright by" usr
0     "ca - verlag gmbh" usr
0     "————————" usr
44  blocks free.
```

DISC 17

```
0  "commodore disc17" 17 2a
1    "boot.64",8,1:     prg
9    "disclader64/128"  prg
0    "————————" usr
1    "eisenbahn",8:     prg
1    "rand&sprite",8:   prg
7    "r&s"              prg
57   "eb"               prg
3    "grafik128-64",8:  prg
2    "p128"             prg
37   "startrotter"      prg
37   "conquest"         prg
37   "e.t."             prg
1    "apfelberge",8:    prg
68   "apfel/c.obj"      prg
35   "assemb.kurs1",8:  prg
4    "mc-routinen 1"    prg
11   "key",8:           prg
0    "————————" usr
24   "check-liste":     prg
4    "menue"            prg
122  "buero 128":       prg
6    "info buero"       seq
6    "info datei"       seq
6    "info kombi"       seq
6    "info count"       seq
6    "info texte"       seq
7    "info seq.file"    seq
22   "stardat":         prg
26   "autogramm.r"      rel
2    "autogramm.s"      seq
93   "lost informat.":  prg
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
23  blocks free.
```

DISC 21

```
0  "commodore disc21" 21 2a
1    "boot.64",8,1:     prg
10   "disclader64/128"  prg
0    "——c 64:——" usr
13   "conv.master",8:   prg
6    "find",8:          prg
17   "schreibschutz",8  prg
57   "raumpatr.",8:     prg
18   "diskformat",8:    prg
10   "diskcopy",8:      prg
0    "——128 pc:——" usr
40   "easypaint":       prg
8    "easymasch"        prg
5    "easypacker v1.0"  prg
9    "grafiklader":     prg
1    "fkopie"           prg
2    "sprites"          prg
1    "sprites2"         prg
2    "demoshapes"       prg
9    "demosatz1"        prg
9    "demosatz2"        prg
9    "demosatz3"        prg
37   "mercedes sel"     prg
37   "pretty woman"     prg
10   "p.mercedes sel"   prg
20   "p.stilleben"      prg
23   "p.tutanchamun"    prg
12   "p.pretty woman"   prg
8    "p.elektronik"     prg
5    "mercedes sel  .f" prg
5    "stilleben     .f" prg
5    "tutanchamun   .f" prg
0    "————————" usr
34   "spritemaster":    prg
16   "anl.spr.master":  prg
3    "spr.mann"         prg
0    "————————" usr
106  "das ding":        prg
0    "————————" usr
35   "maskgenerator":   prg
5    "maskleser":       prg
6    "testprogramm":    prg
9    "mask.beispiel":   seq
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
```

DISC 18

```
0  "commodore disc18" 18 2a
1    "boot.64",8,1:     prg
9    "disclader64/128"  prg
0    "————————" usr
18   "assemb.kurs2",8:  prg
34   "mc-routinen 2"    prg
6    "startadresse",8:  prg
62   "kontakte",8:      prg
48   "space enemy",8:   prg
0    "————————" usr
9    "lotto-kosten":    prg
6    "ausgaben   laden" prg
24   "auswert.  normal" prg
79   "auswert.  system" prg
24   "kosten    normal" prg
29   "kosten    system" prg
15   "spiel    '77'   " prg
32   "zahlenermittlung" prg
135  "super-orion":     prg
75   "le cube":         prg
49   "heizkosten":      prg
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
9  blocks free.
```

DISC 20

```
0  "commodore disc20" 20 2a
1    "boot.64",8,1:     prg
10   "disclader64/128"  prg
0    "————————" usr
21   "assemb.kurs4",8:  prg
9    "mc-routinen 4a"   prg
8    "mc-routinen 4b"   prg
11   "file-kopie",8:    prg
47   "introcoder",8:    prg
32   "demo",8:          prg
60   "dir.-sort",8:     prg
0    "————————" usr
182  "spiel d.lebens":  prg
161  "texas-oil":       prg
32   "titel"            prg
74   "k+g verwaltung":  prg
2    "kopfstand":       prg
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
```

DISC 22

```
0  "commodore disc22" 22 2a
1    "boot.64",8,1:     prg
10   "disclader64/128"  prg
0    "——c 64:——" usr
18   "cd-worktool",8:   prg
4    "quickdisk",8:     prg
8    "sprite spec.",8:  prg
0    "————————" usr
60   "biero-city",8:    prg
1    "bc.help"          prg
152  "bc.main"          prg
0    "——128 pc:——" usr
161  "seewoelfe"        prg
37   "seekarte 3"       prg
3    "sprites 1a"       prg
3    "sprites 2"        prg
3    "sprites 3"        prg
3    "sprites 4"        prg
3    "sprites 5"        prg
3    "sprites 6"        prg
3    "sprites 7"        prg
3    "sprites 8"        prg
3    "sprites 9"        prg
3    "sprites 10"       prg
0    "————————" usr
45   "database pro":    prg
1    ".dmask"           prg
4    ".dbase"           prg
0    "————————" usr
34   "roulette":        prg
10   "2 bildschirme":   prg
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
```

DISC 27

```
0  "commodore disc27" 27 2a
1    "boot.64",8,1      prg
10   "disclader64/128"  prg
0    "————————" usr
0    "     c-64"        usr
0    "————————" usr
73   "etistar",8:       prg
1    "ep blau"          seq
1    "ep epson"         seq
1    "ep normal"        seq
2    "es werbung"       seq
0    "————————" usr
10   "ap64-v1.3",8:     prg
0    "————————" usr
1    "web.loader",8:    prg
13   "web"              prg
7    "eb demo1"         prg
2    "eb demo2"         prg
3    "eb demo3 plot"    prg
2    "eb demo4 lines"   prg
8    "eb demo6 bmtext"  prg
1    "eb demo7 printat" prg
1    "eb flash5"        prg
4    "eb pacmen wait"   prg
3    "1-9 ziff195.4 ff" prg
2    "eb clock show"    prg
30   "eb einweisung"    prg
9    "demo zeisa"       prg
6    "aut.stg.v4",8,1   prg
0    "————————" usr
99   "bigboss",8:       prg
0    "————————" usr
79   "roulette",8:      prg
0    "————————" usr
0    "     c-128"       usr
0    "————————" usr
74   "3d muehle":       prg
0    "————————" usr
56   "kanonendonner":   prg
2    "kd.landschaft"    prg
3    "kd.sprites"       prg
1    "l.standard"       seq
0    "————————" usr
68   "quizmaster":      prg
3    "qm wort-daten"    seq
0    "————————" usr
43   "starvokabel7.0":  prg
4    "demodatei"        seq
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
41  blocks free.
```

DISC 25

```
0  "commodore disc25" 25 2a
1    "boot.64",8,1:     prg
10   "disclader64/128"  prg
0    "——c64:——" usr
55   "assemb.kurs5",8:  prg
1    "mc.ak5"           prg
0    "————————" usr
19   "cpu-trainer",8:   prg
75   "cd-compiler",8:   prg
57   "backgammon",8:    prg
92   "snake",8:         prg
0    "——128 pc:——" usr
17   "diskmon":         prg
13   "screen master":   prg
0    "————————" usr
104  "graph iiic":      prg
33   "gp2"              prg
1    "js2"              prg
2    "hardcopy1500"     prg
2    "hardcopy2"        prg
0    "————————" usr
81   "giro-konto":      prg
3    "12345"            seq
0    "————————" usr
28   "patience":        prg
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
69  blocks free.
```

DISC 24

```
0  "commodore disc24" 24 2a
1    "boot.64",8,1:     prg
10   "disclader64/128"  prg
0    "——c64:——" usr
93   "zauberwald",8:    prg
19   "big copy",8,1:    prg
0    "————————" usr
6    "disk-verz.ii",8:  prg
1    "dv1"              prg
17   "dv2"              prg
72   "dv3"              prg
1    "dv4"              prg
3    "dv5"              prg
3    "dv6"              prg
0    "————————" usr
45   "notus",8:         prg
3    "demo.notus"       prg
0    "————————" usr
3    "helligkeit",8:    prg
0    "——128pc:——" usr
52   "pokermat":        prg
0    "————————" usr
10   "funwriter":       prg
3    "linker":          prg
3    "letwork"          prg
9    "let:big demo"     prg
10   "big demo":        prg
0    "————————" usr
59   "disklabel":       prg
120  "universaldat":    prg
70   "k+g verw.ii":     prg
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
50  blocks free.
```

DISC 28

```
0  "commodore disc28" 28 2a
1    "boot.64",8,1      prg
9    "disclader64/128"  prg
0    "————————" usr
0    "     c-64"        usr
0    "————————" usr
2    "crack"            prg
5    "cracker",8,1      prg
7    "crackerdemo1",8:  prg
6    "crackerdemo2",8:  prg
0    "————————" usr
69   "rus.roulette",8:  prg
9    "rand und sprites" prg
0    "————————" usr
2    "ocp"              prg
1    "show-ocp",8,1     prg
40   "baboon mpic",8,1  prg
0    "————————" usr
70   "main",8:          prg
3    "lieferanten"      usr
14   "waren"            usr
0    "————————" usr
1    "sampler",8,1      prg
0    "————————" usr
0    "     c-128"       usr
0    "————————" usr
1    "psycho-test128":  prg
70   "p"                prg
149  "p-128/1"          prg
123  "p-128/2"          prg
0    "————————" usr
17   "sprite-printer":  prg
3    "sprites"          prg
0    "————————" usr
39   "roem.zeichen":    prg
0    "————————" usr
0    "(c) copyright by" usr
0    "ca - verlag gmbh" usr
0    "————————" usr
22  blocks free.
```

DISC 26

```
0   "commodore disc26" 26 2a
1     "boot.64",8,1       prg
10    "disclader64/128"   prg
0     "————————"  usr
0     "    c-64"          usr
0     "————————"  usr
17    "graph-mon1",8,1    prg
18    "graph-mon2",8,1    prg
0     "————————"  usr
10    "initial",8:        prg
5     "zs"                prg
2     "keytab.byt.o"      prg
15    "main.tx-1.o"       prg
33    "main.tx-2.o"       prg
2     "main.tx-3.o"       prg
2     "t.cent endlos "    prg
2     "t.iec endlos  "    prg
2     "t.cent einzel "    prg
2     "t.iec einzel  "    prg
3     "m.Phrasen     "    prg
3     "m.Cursor      "    prg
20    "treiber-gen."      prg
0     "————————"  usr
137   "usl",8:            prg
0     "————————"  usr
37    "bundesligat.",8:   prg
1     "bayern muenchen"   seq
1     "bor.m'gladbach"    seq
1     "karlsruher sc"     seq
1     "vfb stuttgart"     seq
1     "vfl bochum"        seq
1     "werder bremen"     seq
1     "1.fc nuernberg"    seq
1     "waldhof mannheim"  seq
1     "bayer leverkusen"  seq
1     "1.fc koeln"        seq
1     "bayer uerdingen"   seq
1     "hannover 96"       seq
1     "bor.dortmund"      seq
1     "stuttg.kickers"    seq
1     "hamburger sv"      seq
1     "st.pauli hamburg"  seq
1     "1.fc kaisersl."    seq
1     "eintr.frankfurt"   seq
0     "————————"  usr
0     "    c-128"         usr
0     "————————"  usr
60    "benzinverbr.":     prg
0     "————————"  usr
6     "easy.loader":      prg
10    "easy.basic"        prg
1     "showcom $0b00"     prg
0     "————————"  usr
19    "vitrometer 128":   prg
0     "————————"  usr
62    "dir-master":       prg
0     "————————"  usr
93    "labyrinth":        prg
1     "high-scores-lab."  seq
0     "————————"  usr
16    "zahlenraster":     prg
0     "————————"  usr
0     "(c) copyright by"  usr
0     "ca - verlag gmbh"  usr
0     "————————"  usr
57  blocks free.
```

DISC 29

```
0 "commodore disc29" 29 2a
1     "boot.64",8,1       prg
10    "disclader64/128"   prg
0     "================"  usr
0     "     c-64"         usr
0     "================"  usr
17    "albas",8:          prg
5     "demo-menue"        prg
1     "order 4.6"         prg
1     "order 4.7"         prg
2     "order 4.8"         prg
1     "order 4.9"         prg
3     "order 4.10"        prg
1     "order 4.14"        prg
1     "order 4.15"        prg
0     "————————" usr
2     "skyfox",8:         prg
21    "pctr.lvl"          prg
17    "init.hlp"          prg
10    "main.map"          prg
1     "help.hlp"          prg
0     "————————" usr
53    "light master",8:   prg
2     "demos"             prg
0     "————————" usr
1     "char.loader",8:    prg
46    "chared"            prg
9     "charset1"          prg
9     "charset2"          prg
9     "charset3"          prg
9     "charset4"          prg
9     "charset5"          prg
0     "————————" usr
57    "kreuzungen",8:     prg
3     "m.p. kreuzungen"   prg
1     "h.s. kreuzungen"   usr
21    "lader kreuzungen"  prg
0     "————————" usr
0     "================"  usr
0     "     c-128"        usr
0     "================"  usr
1     "lable.loader":     prg
2     "lable"             prg
3     "demo-lable"        prg
32    "lable.quell"       prg
0     "————————" usr
6     "eddi loder":       prg
40    "zeichen eddi"      prg
9     "demo"              prg
0     "————————" usr
2     "loder":            prg
6     "game box 1"        prg
9     "game box 2"        prg
2     "game box 3"        prg
11    "game box 4"        prg
8     "game box 5"        prg
0     "————————" usr
62    "rubiks clock":     prg
0     "————————" usr
14    "dateiverwalt.":    prg
81    "dv128"             prg
3     "demofile.dat"      seq
0     "================"  usr
0     "(c) copyright by"  usr
0     "ca - verlag gmbh"  usr
0     "================"  usr
```

DISC 30

```
0 "commodore disc30" 30 2a
1     "boot.64"            prg
10    "disclader64/128"    prg
0     "================"   del<
0     "     c-64       "   del<
0     "================"   del<
42    "denga"              prg
0     "————————"  del<
62    "vocab.-master v2"   prg
1     "testvokabeln.vo"    seq
0     "————————"  del<
23    "werkzeuginstruct"   prg
1     "wtool.lader"        prg
6     "work tool"          prg
8     "w.tool.o"           prg
0     "————————"  del<
92    "chr$generator"      prg
4     "lader.basic"        prg
15    "gesicht"            prg
9     "labyrinth"          prg
15    "haeschen"           prg
0     "================"   del<
0     "c64/c128/c16/p4 "   del<
0     "================"   del<
7     "datawandler.uni"    prg
0     "================"   del<
0     "     c-128      "   del<
0     "================"   del<
76    "multi-mathe"        prg
0     "————————"  del<
156   "daten.rex"          prg
1     "0d00"               prg
1     "1400"               prg
1     "1300"               prg
0     "————————"  del<
28    "memory"             prg
0     "————————"  del<
39    "quattro"            prg
33    "q3-graphic4"        prg
0     "————————"  del<
0     "(c) copyright by"   del<
0     "ca - verlag gmbh"   del<
0     "————————"  del<
```

DISC 32

```
0 "commodore disc32" 32 2a
1     "boot.64"            prg
10    "disclader64/128"    prg
0     "================"   del<
0     "     c-64       "   del<
0     "================"   del<
6     "startmenue.c64"     prg
14    "alkoholkontrolle"   prg
0     "————————"  del<
54    "komplan v2.0"       prg
13    "komplan.masch"      prg
1     "komplan.kuerzel"    prg
1     "komplan.zeiten"     prg
3     "torsten/12.1.st"    prg
3     "ralf/13.1.st"       prg
3     "ralf/13.2.st"       prg
3     "future.st"          prg
0     "————————"  del<
6     "berti ii"           prg
9     "test......$c000"    prg
11    "high-score.$9300"   prg
1     "h-s.regist.$9e00"   prg
12    "menue+help.$7000"   prg
2     "bonus......$7e00"   prg
1     "char-copy..$8000"   prg
1     "platz......$804c"   prg
1     "bildrechts.$80ed"   prg
1     "punkte aus.$81a5"   prg
1     "zeitbalken.$824f"   prg
1     "bild.holen.$8318"   prg
1     "timeout....$839c"   prg
2     "stor.close.$8440"   prg
4     "laufrout...$8571"   prg
2     "warte.joy2.$886d"   prg
2     "lader.1+s..$8a07"   prg
1     "sound......$8bfd"   prg
1     "levelcop...$8cd0"   prg
1     "labild.lev.$8d55"   prg
1     "labild.gam.$8e4a"   prg
2     "new-char.1.$9000"   prg
10    "sprite.02..$f000"   prg
1     "level  1"           prg
1     "level  2"           prg
1     "level  3"           prg
1     "level  4"           prg
1     "level  5"           prg
1     "level  6"           prg
1     "level  7"           prg
1     "level  8"           prg
1     "level  9"           prg
1     "level 10"           prg
33    "screen  1"          prg
33    "screen  2"          prg
33    "screen  3"          prg
33    "screen  4"          prg
33    "screen  5"          prg
0     "================"   del<
0     "     c-128      "   del<
0     "================"   del<
6     "startmenue.c128"    prg
43    "kasino"             prg
14    "family-quiz"        prg
0     "————————"  del<
7     "golf-designer"      prg
10    "golf-player"        prg
0     "————————"  del<
52    "diathek a"          prg
52    "diathek a/b"        prg
0     "————————"  del<
9     "hires-info"         prg
2     "hirestest"          prg
3     "matrixkonvert"      prg
4     "hires.cbmed"        prg
4     "hires.epsed"        prg
4     "hires.ibmed"        prg
5     "hires.cbmgz"        prg
4     "hires.epscd"        prg
4     "hires.cbmdd"        prg
4     "hires.epstd"        prg
4     "hires.ibmtd"        prg
5     "hires.epsrv"        prg
5     "hires.ibmrv"        prg
5     "hires.ibmepshd"     prg
5     "hires.ibmepshrv"    prg
2     "centronic.c128gr"   prg
0     "————————"  del<
8     "hubschrauber"       prg
18    "garfield"           prg
0     "————————"  del<
```


DISC 31

```
0 "commodore disc31" 31 2a
1     "boot.64"            prg
10    "disclader64/128"    prg
0     "================"   del<
0     "     c-64       "   del<
0     "================"   del<
3     "key.64"             prg
0     "————————"  del<
2     "sprite_demo"        prg
3     "sd_1"               prg
12    "sd_2"               prg
10    "sd_3"               prg
0     "————————"  del<
1     "sprite_tool"        prg
3     "s.tool.obj"         prg
13    "s.tool.basic"       prg
27    "s.tool.ass"         prg
0     "————————"  del<
40    "gelbe kugeln.64"    prg
7     "mp gelbe kugeln"    prg
0     "————————"  del<
50    "sprint"             prg
0     "================"   del<
0     "     c-128      "   del<
0     "================"   del<
6     "cross"              prg
0     "————————"  del<
25    "chardefine 128"     prg
23    "editor .obj"        prg
9     "Z'zeichensatz1"     prg
9     "Z'zeichensatz2"     prg
0     "————————"  del<
31    "tabellenkalku"      prg
14    "tab.test1"          seq
13    "tab.test2"          seq
0     "————————"  del<
1     "adrload"            prg
3     "aspr"               prg
23    "address v2.0"       prg
0     "————————"  del<
2     "pic-gallery v3.7"   prg
37    "written in 1988 "   prg
74    "      by        "   prg
37    "  uwe schwesig  "   prg
33    "hires-bild 1"       prg
32    "hires-bild 2"       prg
0     "————————"  del<
6     "eingabefeld.40"     prg
6     "eingabefeld.80"     prg
9     "datenmaske.40"      prg
9     "datenmaske.80"      prg
31    "textedi"            prg
0     "————————"  del<
33    "star paint"         prg
0     "================"   del<
0     "(c) copyright by"   del<
0     "ca - verlag gmbh"   del<
0     "================"   del<
15 blocks free.
```

DISC 33

```
0 "commodore disc33" 33 2a
1    "boot.64"           prg
10   "disclader64/128"   prg
0    "================"  del<
0    "     c-64       "  del<
0    "================"  del<
6    "dcopy64"           prg
32   "puzzle"            prg
5    "mp puzzle"         prg
28   "zahlenraetsel"     prg
0    "----------" del<
28   "super racer"       prg
1    "hi-score"          seq
4    "lev 1"             seq
6    "lev 2"             seq
0    "----------" del<
20   "info"              prg
27   "loader fuer"       prg
1    "tief"              seq
138  "im kerker"         prg
2    "gefangen"          prg
0    "================"  del<
0    "     c-128      "  del<
0    "================"  del<
14   "painter"           prg
0    "----------" del<
6    "hires info"        prg
5    "hiresdb.eps"       prg
5    "hiresdb.epstd"     prg
5    "hiresdb.ibmtd"     prg
5    "hiresdb.ibmepshd"  prg
6    "hiresdb.cbmgz"     prg
2    "centronic.c128gr"  prg
0    "----------" del<
1    "z80-mon.lader"     prg
30   "z80-mon 1.0"       prg
1    "z80-svc.lader"     prg
9    "z80-svc 1.0"       prg
0    "----------" del<
76   "secondhand v2.0"   prg
0    "----------" del<
40   "a.d.i.v1.0/128er"  prg
1    "m1:beispiel"       seq
1    "m2:beispiel"       seq
1    "d1:beispiel"       seq
1    "d2:beispiel"       seq
0    "----------" del<
33   "sort 128"          prg
83   "bruchtrainer"      prg
3    "input korrektur"   prg
0    "================"  del<
0    "(c) copyright by"  del<
0    "ca - verlag gmbh"  del<
0    "================"  del<
26 blocks free.
```

DISC 34

```
0 "commodore disc34" 34 2a
1    "boot.64"           prg
10   "disclader64/128"   prg
0    "================"  del<
0    "     c-64       "  del<
0    "================"  del<
72   "labyrinth"         prg
54   "tapemaster"        prg
63   "compmaster"        prg
0    "----------" del<
1    "gnampf...!"        prg
9    "levls"             prg
29   "edi"               prg
48   "main"              prg
32   "pic"               prg
9    "load"              prg
0    "----------" del<
1    "sidlader"          prg
9    "sid-mon"           prg
1    "upmlader"          prg
5    "upm"               prg
42   "bill-manager"      prg
0    "================"  del<
0    "     c-128      "  del<
0    "================"  del<
17   "geographics"       prg
27   "niedersachsen"     seq
0    "----------" del<
3    "lies mich"         prg
1    "graphic-designer"  prg
1    "gd1"               prg
2    "gd2"               prg
1    "gd3"               prg
3    "gd4"               prg
77   "gd5"               prg
5    "gd6"               prg
5    "z-normal (ascii)"  prg
5    "z-din"             prg
5    "z-multicolor 1"    prg
5    "z-multicolor 2"    prg
5    "z-multicolor 3"    prg
9    "z-kl.western"      prg
9    "z-esprit"          prg
9    "z-verzerrt"        prg
9    "z-computer"        prg
9    "z-block"           prg
0    "----------" del<
17   "calc.prg.ver.1.0"  prg
1    "hardcopy80.$1300"  prg
4    "test"              seq
0    "----------" del<
21   "lgs 128"           prg
22   "kryptogramm 128"   prg
0    "================"  del<
0    "(c) copyright by"  del<
0    "ca - verlag gmbh"  del<
0    "================"  del<
```

Impressum

Commodore-Disc erscheint monatlich in der CA- Verlags GmbH, Heßstraße90, D-8000 München 40
Tel.: 089/1298011, Telex: 5214428 cav-d

VERANTWORTLICH FÜR DEN INHALT:
Alfons Mittelmeyer

REDAKTION & STÄNDIGE MITARBEITER:
T.Seibt, A.Herman, A.Mittelmeyer, M.Gersteler

GESCHÄFTSFÜHRER:
Werner E. Seibt

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN:
Leserservice, Postf. 1161,
8044 Unterschleißheim

VERTRIEB:
VPM Wiesbaden

Printed in Germany by ADV,
D-8900 Augsburg 1
Aindlingerstraße 17-19

Jugend forscht

Mit dem Computer dabeisein

Mit einem C128, einem Fischertechnik-Baukasten und einem Computerinterface derselben Firma gewannen Reinhard Holler und Uwe Rebhahn bei JUGEND FORSCHT 89.

Mitmachen und gewinnen heißt die Device. Das dachten sich Reinhard Holler und Uwe Rebhahn aus München, und das im Alter von dreizehn und zwölf Jahren. Die Idee dazu hatte Uwe Rebhahn, der sie seinem Freund Reinhard, Commodore 128- und Fischertechnik-Baukasten-Besitzer, vorschlug. Uwes Vorstellung war, ein Wertgegenstand sollte derart geschützt werden, daß er bei Unterbrechung einer Lichtschranke diebstahlsicher abgesenkt werde. Dies könne in einem Turm untergebracht werden, gebaut aus Fischertechnik-Baukastenteilen.

Reinhard machte sich gleich daran, das geplante Modell zu basteln. Auf einer Grundplatte erhebt sich, etwa zehn Zentimeter breit und sieben Zentimeter tief, der etwa zwanzig Zentimeter hohe Turm aus Plasik- und Aluminiumteilen. Ein Tablett für einen kleinen Wertgegenstand, etwa eine Geldmünze, ist im Turminnern an einer Zahnstange senkrecht fahrbar mit Hilfe eines Elektro-Motors eingebaut. Eine Lichtschranke soll nähernde Hände registrieren und das Absenken des Wertgegenstandes veranlassen.
Es braucht hierzu nicht nur eine Stromversorgung, sondern auch eine entsprechende Steuereinrichtung. Wo es an Elektronikkenntnissen mangelt, ist der Computer der Retter in der Not. Mit dem Fischertechnik-Computerinterface und dem C128 mangelt es nicht an der erforderlichen Hardware.
Der im Interface eingebaute Analog-Digitalwandler übergibt den Istwert der Lichtstärke an den Computer. Zu vergleichen damit ist die Soll-Lichtstärke. Fällt der Istwert unter einen unzulässigen Wert, so liegt eine Unterbrechung der Lichtschranke vor. In diesem Fall ist das Relais im Interface zu schließen, so daß der Motor das Tragetablett nach unten fährt.

Nach einigen Verbesserungen am Modell und der Bedienerführung konnte man zufrieden sein. Für JUGEND FORSCHT brauchte es nur noch eine schriftliche Darstellung.
Reinhard und Uwe meinen, daß sich die Mühe gelohnt habe. ■

NACHBESTELLUNG

❒ 01 ❒ 02 ❒ 03 ❒ 04 ❒ 05 ❒ 06 ❒ 07 ❒ 08 ❒ 09 ❒ 10 ❒ 11
❒ 12 ❒ 13 ❒ 14 ❒ 15 ❒ 16 ❒ 17 ❒ 18 ❒ 19 ❒ 20 ❒ 21 ❒ 22
❒ 23 ❒ 24 ❒ 25 ❒ 26 ❒ 27 ❒ 28 ❒ 29 ❒ 30 ❒ 31 ❒ 32 ❒ 33

Hiermit bestelle ich die angekreuzten Exemplare der COMMODORE DISC:

Preis pro Diskette + Computermagazin: DM 19,80
ACHTUNG SONDERANGEBOT: 12 Ausgaben für nur DM 200.- (incl. Versand)
Versandkostenanteil, pro Lieferung: DM 2,20
Senden Sie bitte die Diskette an

Name ____________________ **Vorname** ____________________

Straße/Nr. ____________________ **PLZ/Ort** ____________________

Ich zahle (Zutreffendes ankreuzen)
❒ **per Scheck** ❒ **Bar -Geld liegt bei** ❒ **per Bankabbuchung am Versandtag (nur in der BRD möglich!)**

Meine Bank (Name) ____________________ **Meine Kontonummer** ____________________

Bankleitzahl ____________________ **(steht auf jedem Scheck)**

Unterschrift des Bestellers ____________________

Ausschneiden und einsenden (oder fotokopiert an) ✏ COMMODORE DISC, Postfach 1161, D-8044 Unterschleißheim

✂ -

SPAREN SIE 40 MARK!
COMMODORE DISC JETZT IM ABO

ABO-SERVICE COUPON

(Gültig nur innerhalb der Bundesrepublik Deutschland und Westberlin)
Ja, ich möchte von Ihrem Angebot Gebrauch machen. Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten zwölf Ausgaben an untenstehende Anschrift. Wenn ich nicht vier Wochen vor Ablauf kündige, läuft diese Abmachung automatisch weiter.
WICHTIG: Sie können diesen Auftrag binnen einer Woche nach Zugang der Abo-Bestätigung widerrufen! Es genügt die rechtzeitige Absendung an den Verlag.

Name: ____________________

Vorname: ____________________

Straße/Nr.: ____________________

PLZ/Ort: ____________________

Ich bezahle DM 200,- (inkl.Mehrwertsteuer) statt DM 237,60 für die nächsten 12 Ausgaben
O per beiliegenden Verrechnungs-/Euroscheck
O per Rechnung
O bargeldlos per Bankeinzug von meinem Konto bei

Bank/Ort: ____________________

Kontonummer: ____________________

Bankleitzahl: ____________________
(steht auf jedem Kontoauszug)

Unterschrift: ____________________

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.
2. Unterschrift: ____________________

COMMODORE DISC
ABO-SERVICE 35
Postfach 1161
D-8044 Unterschleißheim